



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

ETSEIB/GIE

7 de novembre de 2022

1 Conceptes

2 Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

3 Tipus de grafs

4 Recorregut en primera profunditat

5 Recorregut en primera amplada

6 Exercici: Camí més curt entre dos vèrtexs

7 Exercici: Camí menys costós entre dos vèrtexs

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

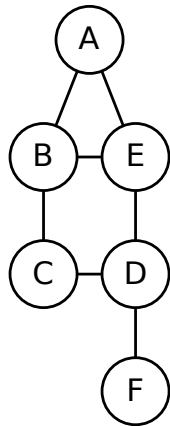
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Un **graf** és una representació d'un conjunt d'objectes en que alguns parells d'objectes es connecten mitjançant enllaços.
- La representació d'un objecte l'anomenem **vèrtex** o **node**
- La representació de l'enllaç que connecta dos vèrtexs l'anomenem **aresta** o **arc**.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

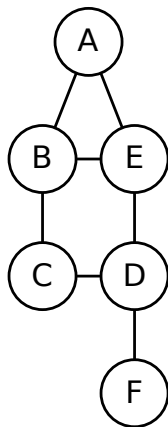
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Un **graf** és una representació d'un conjunt d'objectes en que alguns parells d'objectes es connecten mitjançant enllaços.
- La representació d'un objecte l'anomenem **vèrtex** o **node**
- La representació de l'enllaç que connecta dos vèrtexs l'anomenem **aresta** o **arc**.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

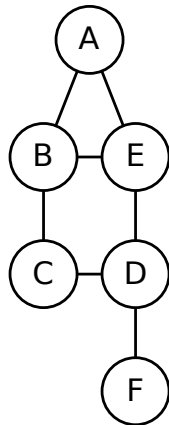
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Un **graf** és una representació d'un conjunt d'objectes en que alguns parells d'objectes es connecten mitjançant enllaços.
- La representació d'un objecte l'anomenem **vèrtex** o **node**
- La representació de l'enllaç que connecta dos vèrtexs l'anomenem **aresta** o **arc**.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

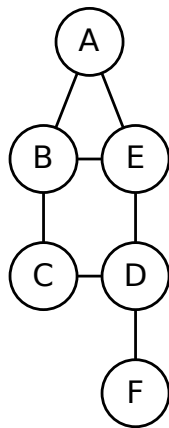
Exercici: Camí menys curt entre

- Més formalment, un graf és un parell ordenat $G = (V, E)$ que comprèn un conjunt V de vèrtexs, i un conjunt E d'arestes. Una aresta és una parella d'objectes de V .

- $V = \{A, B, C, D, E, F\}$

-

$$E = \{(A, B), (A, E), (C, B), (C, D), (B, E), (E, D), (D, F)\}$$



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

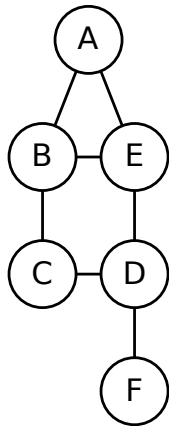
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

- Més formalment, un graf és un parell ordenat $G = (V, E)$ que comprèn un conjunt V de vèrtexs, i un conjunt E d'arestes. Una arista és una parella d'objectes de V .
- $V = \{A, B, C, D, E, F\}$

$$E = \{(A, B), (A, E), (C, B), (C, D), (B, E), (E, D), (D, F)\}$$



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

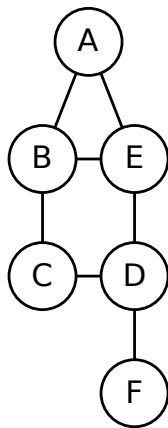
Exercici: Camí menys curt entre

- Més formalment, un graf és un parell ordenat $G = (V, E)$ que comprèn un conjunt V de vèrtexs, i un conjunt E d'arestes. Una arista és una parella d'objectes de V .

- $V = \{A, B, C, D, E, F\}$

-

$$E = \{(A, B), (A, E), (C, B), (C, D), (B, E), (E, D), (D, F)\}$$



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

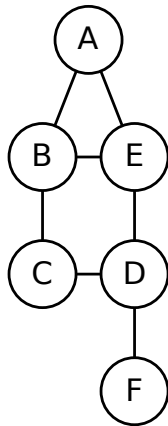
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- L'**ordre** d'un graf: $|V|$ (nombre de vèrtexs)
6 vèrtexs en l'exemple.
- La **mida** d'un graf: $|E|$ (nombre d'arestes)
7 arestes en l'exemple.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

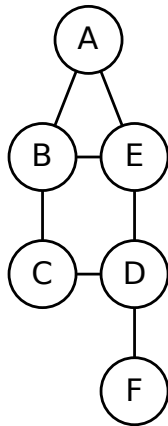
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre dos vèrtexs

- L'**ordre** d'un graf: $|V|$ (nombre de vèrtexs)
6 vèrtexs en l'exemple.
- La **mida** d'un graf: $|E|$ (nombre d'arestes)
7 arestes en l'exemple.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

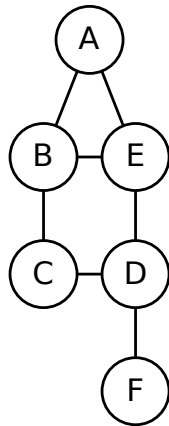
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- L'**ordre** d'un graf: $|V|$ (nombre de vèrtexs)
6 vèrtexs en l'exemple.
- La **mida** d'un graf: $|E|$ (nombre d'arestes)
7 arestes en l'exemple.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

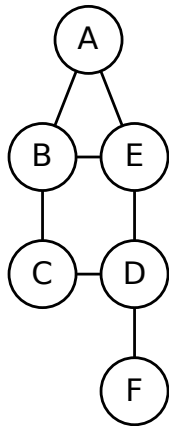
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- L'**ordre** d'un graf: $|V|$ (nombre de vèrtexs)
6 vèrtexs en l'exemple.
- La **mida** d'un graf: $|E|$ (nombre d'arestes)
7 arestes en l'exemple.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

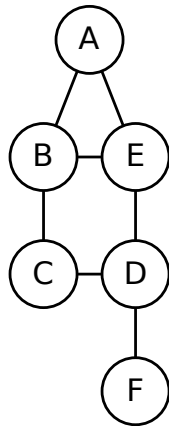
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre dos vèrtexs

- Un **camí** és una seqüència d'arestes que connecten una seqüència de vèrtexs.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

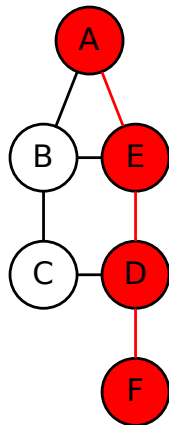
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

- Un **camí** és una seqüència d'arestes que connecten una seqüència de vèrtexs.
- Camí AEDF



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

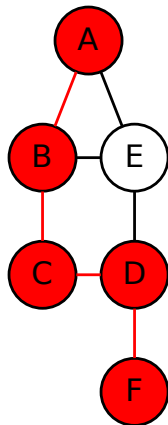
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

- Un **camí** és una seqüència d'arestes que connecten una seqüència de vèrtexs.
- Camí ABCDF



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

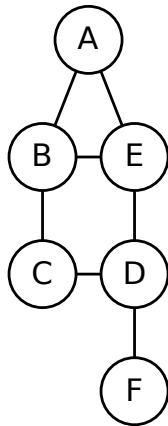
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Un **cicle** és un camí on el seu vèrtex inicial i el seu vèrtex final són el mateix.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

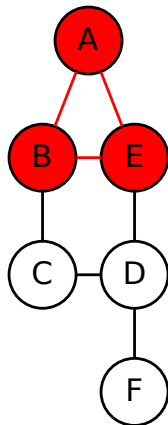
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Un **cicle** és un camí on el seu vèrtex inicial i el seu vèrtex final són el mateix.
- Cicle ABE



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

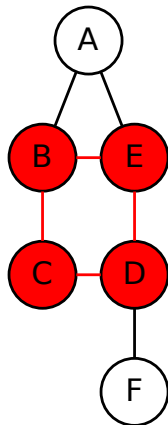
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Un **cicle** és un camí on el seu vèrtex inicial i el seu vèrtex final són el mateix.
- Cicle BEDC



Grafs

Conceptes

Introducció al
networkx

Grafs simètric (no
dirigits)

Camins

Components

Tipus de grafs

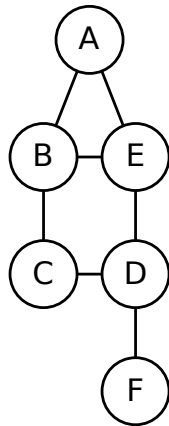
Recorregut en
primera
profunditat

Recorregut en
primera
amplada

Exercici:
Camí més
curt entre dos
vèrtexs

Exercici:
Camí menys
costós entre

- El **grau** d'un vèrtex és el nombre d'arestes que hi connecten. Un vèrtex que es connecta a sí mateix (cicle) compta per dos.
-



Grafs

Conceptes

Introducció al
networkx

Grafs simètric (no
dirigits)

Camins

Components

Tipus de grafs

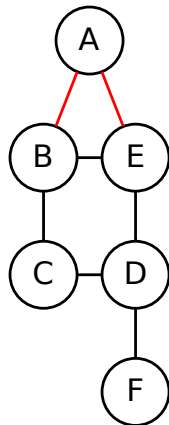
Recorregut en
primera
profunditat

Recorregut en
primera
amplada

Exercici:
Camí més
curt entre dos
vèrtexs

Exercici:
Camí menys
costós entre

- El **grau** d'un vèrtex és el nombre d'arestes que hi connecten. Un vèrtex que es connecta a sí mateix (cicle) compta per dos.
- El grau d'*A* és 2



Grafs

Conceptes

Introducció al
networkx

Grafs simètric (no
dirigits)

Camins

Components

Tipus de grafs

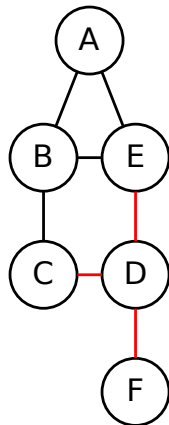
Recorregut en
primera
profunditat

Recorregut en
primera
amplada

Exercici:
Camí més
curt entre dos
vèrtexs

Exercici:
Camí menys
costós entre

- El **grau** d'un vèrtex és el nombre d'arestes que hi connecten. Un vèrtex que es connecta a sí mateix (cicle) compta per dos.
- El grau d' D és 3



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

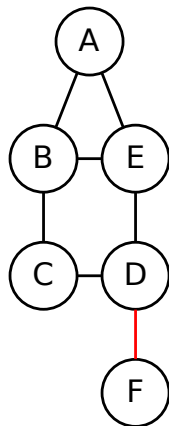
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

- El **grau** d'un vèrtex és el nombre d'arestes que hi connecten. Un vèrtex que es connecta a sí mateix (cicle) compta per dos.
- El grau d' F és 1



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

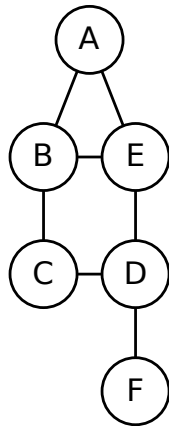
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre dos vèrtexs

- Els **veïns** (vèrtexs adjacents o incidents) de v : vèrtexs units per una aresta a v .



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

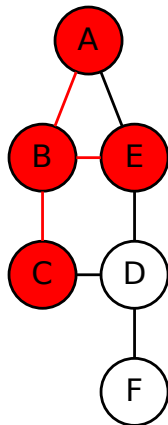
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre dos vèrtexs

- Els **veïns** (vèrtexs adjacents o incidents) de v : vèrtexs units per una aresta a v .
- Els veïns de B són A , E , i C



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

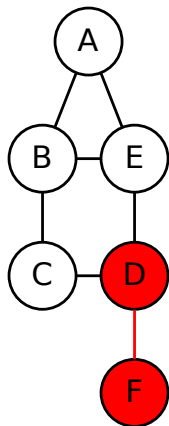
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- Els **veïns** (vèrtexs adjacents o incidents) de v : vèrtexs units per una aresta a v .
- El veí de F és D





networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```

Graf

Conceptes

Introducció al networkx

Graf simètric (no dirigits)

Camins

Components

Tipus de graf

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([(('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F'))])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```

Graf

Conceptes

Introducció al networkx

Graf simètric (no dirigits)

Camins

Components

Tipus de graf

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```

Graf

Conceptes

Introducció al networkx

Graf simètric (no dirigits)

Camins

Components

Tipus de graf

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([(('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F'))])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```



networkx: graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g, ['F','D','E','A','B','C','D'])
>>> g.add_edge('B','E')
>>> g.nodes()
NodeView(('A', 'C', 'B', 'E', 'D', 'F'))
>>> g.edges()
EdgeView([('A', 'B'), ('A', 'E'), ('C', 'B'), ('C', 'D'), ('B', 'E'), ('E', 'D'), ('D', 'F')])
>>> import matplotlib.pyplot as plt
>>> nx.draw(g, with_labels=True)
>>> plt.show()
```

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

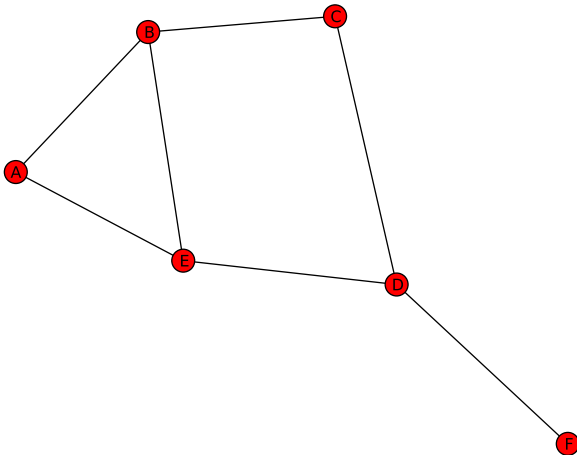
Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

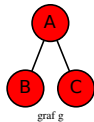
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

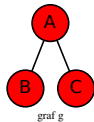
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a'), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

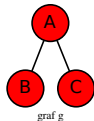
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a'), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

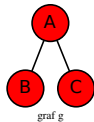
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a'), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

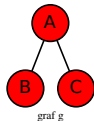
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

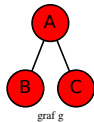
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

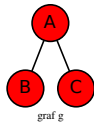
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g
<networkx.classes.graph.Graph object at ...>
>>> g.add_nodes_from(('a', 'b', 'c'))
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a'), ('c', 'a')))
```

En aquest graf, al tenir tots els nodes connectats, no cal cridar `add_nodes_from`. Entrant només les arestes donem d'alta tots els nodes del graf.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

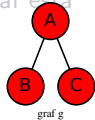
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a'))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

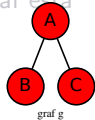
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a'))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



graf g

Consulta d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

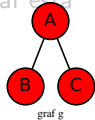
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a'), ('c', 'a'))))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

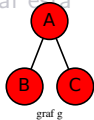
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a'))))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

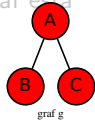
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a'))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

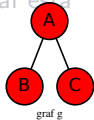
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a'), ('c', 'a'))))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

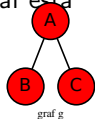
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> g.add_edges_from(((('a', 'b'), ('a', 'c')), ('b', 'a')
), ('c', 'a'))))
>>> g.add_node('b')
>>> g.add_node('b')
>>> sorted(g.nodes())
['a', 'b', 'c']
>>> g.edges()
EdgeView([('a', 'b'), ('a', 'c')])
```

Noteu que hem intentat posar dos cops quatre parelles de nodes i finalment tenim dues arestes. El mateix passa si insistim en posar un mateix node. L'ordre de les parelles depèn de l'execució. El graf està basat en diccionaris.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

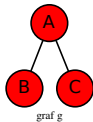
Exercici: Camí menys costós entre

Consulta del node a

```
>>> g['a']  
AtlasView({'b': {}, 'c': {}})
```

Consulta de l'aresta ('a', 'b')

```
>>> g['a']['b']  
{}
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

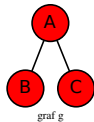
Exercici: Camí menys costós entre

Consulta del node a

```
>>> g['a']  
AtlasView({'b': {}, 'c': {}})
```

Consulta de l'aresta ('a', 'b')

```
>>> g['a']['b']  
{}
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

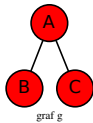
Exercici: Camí menys costós entre

Consulta del node a

```
>>> g['a']  
AtlasView({'b': {}, 'c': {}})
```

Consulta de l'aresta ('a', 'b')

```
>>> g['a']['b']  
{}
```



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

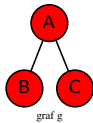
Exercici: Camí menys costós entre

Consulta del node a

```
>>> g['a']  
AtlasView({'b': {}, 'c': {}})
```

Consulta de l'aresta ('a', 'b')

```
>>> g['a']['b']  
{}
```



graf g

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

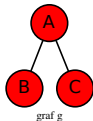
Exercici: Camí menys costós entre

Consulta del node a

```
>>> g['a']  
AtlasView({'b': {}, 'c': {}})
```

Consulta de l'aresta ('a', 'b')

```
>>> g['a']['b']  
{}
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

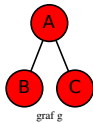
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

`iter(g)` retorna un iterador de tots els nodes del graf

```
>>> iter(g)
<dict_keyiterator object at ...>
>>> for node in g:
...     print(node)
a
b
c
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

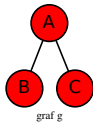
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

`iter(g)` retorna un iterador de tots els nodes del graf

```
>>> iter(g)
<dict_keyiterator object at ...>
>>> for node in g:
...     print(node)
a
b
c
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

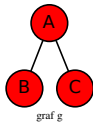
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

`iter(g)` retorna un iterador de tots els nodes del graf

```
>>> iter(g)
<dict_keyiterator object at ...>
>>> for node in g:
...     print(node)
a
b
c
```



Recorregut de nodes del graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

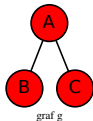
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

`iter(g)` retorna un iterador de tots els nodes del graf

```
>>> iter(g)
<dict_keyiterator object at ...>
>>> for node in g:
...     print(node)
a
b
c
```



graf g

Veïns d'un node del graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

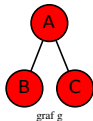
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> for node in g['b']:  
...     print(node)  
a  
  
>>> for node in g.neighbors('b'):  
...     print(node)  
a
```



Veïns d'un node del graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

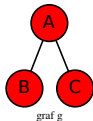
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> for node in g['b']:  
...     print(node)  
a  
  
>>> for node in g.neighbors('b'):  
...     print(node)  
a
```



Veïns d'un node del graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

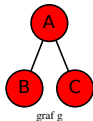
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> for node in g['b']:  
...     print(node)  
a  
  
>>> for node in g.neighbors('b'):  
...     print(node)  
a
```



Veïns d'un node del graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

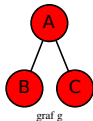
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

```
>>> for node in g['b']:  
...     print(node)  
a  
  
>>> for node in g.neighbors('b'):  
...     print(node)  
a
```



Etiquetes o atributs d'una aresta

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

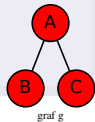
Exercici: Camí menys costós entre

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a','b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([(('a', 'b'), ('a', 'c'))])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



Etiquetes o atributs d'una aresta

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

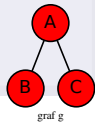
Exercici: Camí menys costós entre

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a','b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([(('a', 'b'), ('a', 'c'))])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



Etiquetes o atributs d'una aresta

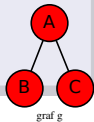
Grafs

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a', 'b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([(('a', 'b'), ('a', 'c'))])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



graf g

Etiquetes o atributs d'una aresta

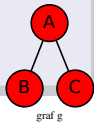
Grafs

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a', 'b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([(('a', 'b'), ('a', 'c'))])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



graf g

Etiquetes o atributs d'una aresta

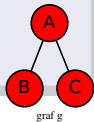
Grafs

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a','b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([(('a', 'b'), ('a', 'c'))])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



graf g

Etiquetes o atributs d'una aresta

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

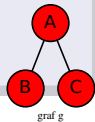
Exercici: Camí menys costós entre

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a','b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([( 'a', 'b'), ( 'a', 'c')])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



graf g

Etiquetes o atributs d'una aresta

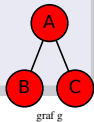
Grafs

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a', 'b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([( 'a', 'b'), ( 'a', 'c')])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



Etiquetes o atributs d'una aresta

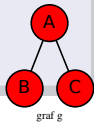
Grafs

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a','b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([(('a', 'b'), ('a', 'c'))])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



Etiquetes o atributs d'una aresta

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

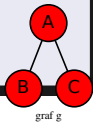
Exercici: Camí menys costós entre

afegim una etiqueta (atribut) *nom* a l'aresta ('a','b')

```
>>> g.add_edge('a','b', nom='ab')
```

El conjunt d'arestes queda igual

```
>>> g.edges()
EdgeView([( 'a', 'b'), ( 'a', 'c')])
>>> g['a']['b']
{'nom': 'ab'}
>>> g['a']['b']['nom']
'ab'
>>> g['b']['a']
{'nom': 'ab'}
>>> g['b']['a']['nom']
'ab'
```



graf g

Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

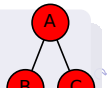
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```



Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

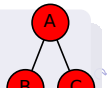
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```



Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

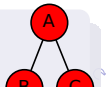
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```



Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

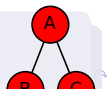
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```





Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

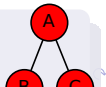
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```





Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

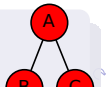
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```





Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

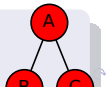
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```





Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

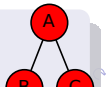
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```





Més consultes d'un graf

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Nombre d'arestes del graf (mida d'un graf)

```
>>> g.size()
2
```

Nombre de nodes del graf (ordre d'un graf)

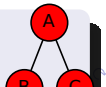
```
>>> len(g)
3
```

Grau d'un node

```
>>> g.degree('a')
2
```

Graus de tots els nodes del graf

```
>>> g.degree()
DegreeView({'a': 2, 'b': 1, 'c': 1})
```



Camins: Creació d'un graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```

Camins: Creació d'un graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```

Camins: Creació d'un graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```

Camins: Creació d'un graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```

Camins: Creació d'un graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

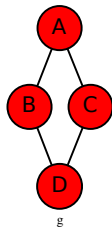
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```



Camins: Creació d'un graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

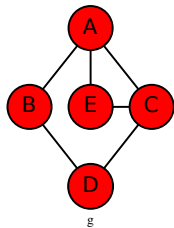
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```

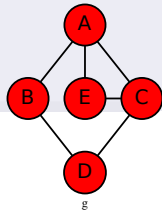


Consulem si hi ha camí entre 'a' i 'd':

```
>>> nx.has_path(g, 'a', 'd')
True
```

Consulem tots els camins simples entre 'a' i 'd'

```
>>> for c in sorted(nx.all_simple_paths(g, 'a', 'd')):
...     print(c)
['a', 'b', 'd']
['a', 'c', 'd']
['a', 'e', 'c', 'd']
```

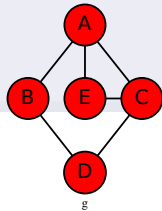


Consulem si hi ha camí entre 'a' i 'd':

```
>>> nx.has_path(g, 'a', 'd')
True
```

Consulem tots els camins simples entre 'a' i 'd'

```
>>> for c in sorted(nx.all_simple_paths(g, 'a', 'd')):
...     print(c)
['a', 'b', 'd']
['a', 'c', 'd']
['a', 'e', 'c', 'd']
```

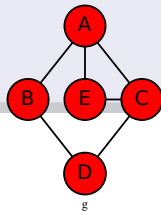


Consulem si hi ha camí entre 'a' i 'd':

```
>>> nx.has_path(g, 'a', 'd')
True
```

Consulem tots els camins simples entre 'a' i 'd'

```
>>> for c in sorted(nx.all_simple_paths(g, 'a', 'd')):
...     print(c)
['a', 'b', 'd']
['a', 'c', 'd']
['a', 'e', 'c', 'd']
```

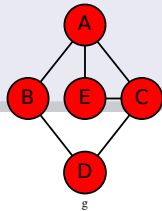


Consulem si hi ha camí entre 'a' i 'd':

```
>>> nx.has_path(g, 'a', 'd')  
True
```

Consulem tots els camins simples entre 'a' i 'd'

```
>>> for c in sorted(nx.all_simple_paths(g, 'a', 'd'  
'')):  
...     print(c)  
['a', 'b', 'd']  
['a', 'c', 'd']  
['a', 'e', 'c', 'd']
```

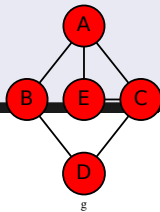


Consulem si hi ha camí entre 'a' i 'd':

```
>>> nx.has_path(g, 'a', 'd')
True
```

Consulem tots els camins simples entre 'a' i 'd'

```
>>> for c in sorted(nx.all_simple_paths(g, 'a', 'd')):
...     print(c)
['a', 'b', 'd']
['a', 'c', 'd']
['a', 'e', 'c', 'd']
```



Consultem la longitud del camí mínim

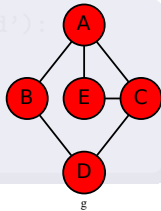
```
>>> nx.shortest_path_length(g, 'a', 'd')
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')
['a', 'b', 'd']
```

Consultem tots els camins mínims

```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):
...     print(c)
...
['a', 'b', 'd']
['a', 'c', 'd']
```



Consultem la longitud del camí mínim

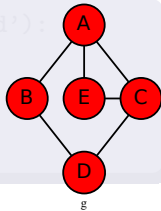
```
>>> nx.shortest_path_length(g, 'a', 'd')
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')
['a', 'b', 'd']
```

Consultem tots els camins mínims

```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):
...     print(c)
...
['a', 'b', 'd']
['a', 'c', 'd']
```



Consultem la longitud del camí mínim

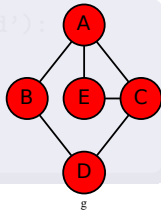
```
>>> nx.shortest_path_length(g, 'a', 'd')  
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')  
['a', 'b', 'd']
```

Consultem tots els camins mínims

```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):  
...     print(c)  
...  
['a', 'b', 'd']  
['a', 'c', 'd']
```



Consultem la longitud del camí mínim

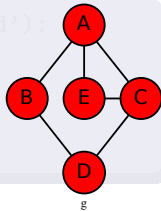
```
>>> nx.shortest_path_length(g, 'a', 'd')  
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')  
['a', 'b', 'd']
```

Consultem tots els camins mínims

```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):  
...     print(c)  
...  
['a', 'b', 'd']  
['a', 'c', 'd']
```



Consultem la longitud del camí mínim

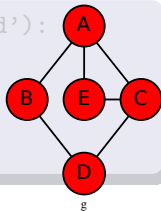
```
>>> nx.shortest_path_length(g, 'a', 'd')
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')
['a', 'b', 'd']
```

Consultem tots els camins mínims

```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):
...     print(c)
...
['a', 'b', 'd']
['a', 'c', 'd']
```



Consultem la longitud del camí mínim

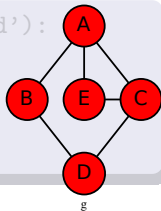
```
>>> nx.shortest_path_length(g, 'a', 'd')
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')
['a', 'b', 'd']
```

Consultem tots els camins mínims

```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):
...     print(c)
...
['a', 'b', 'd']
['a', 'c', 'd']
```



Consultem la longitud del camí mínim

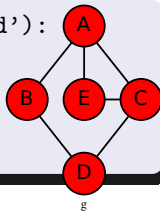
```
>>> nx.shortest_path_length(g, 'a', 'd')
2
```

Consultem un camí mínim

```
>>> nx.shortest_path(g, 'a', 'd')
['a', 'b', 'd']
```

Consultem tots els camins mínims

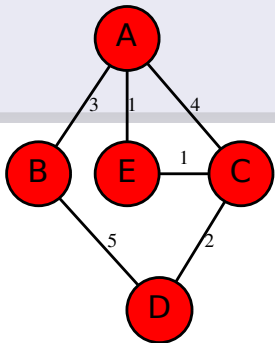
```
>>> for c in nx.all_shortest_paths(g, 'a', 'd'):
...     print(c)
...
['a', 'b', 'd']
['a', 'c', 'd']
```



Afegim un atribut distància a les arestes

```

>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
    
```

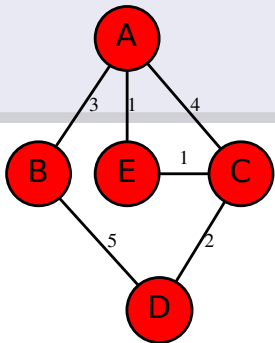


graf g

Afegim un atribut distància a les arestes

```

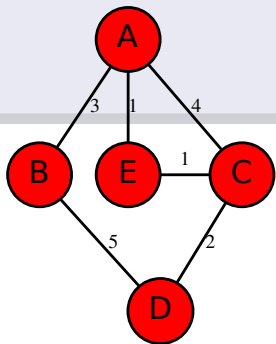
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
    
```



graf g

Afegim un atribut distància a les arestes

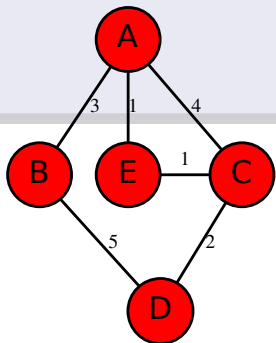
```
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
```



graf g

Afegim un atribut distància a les arestes

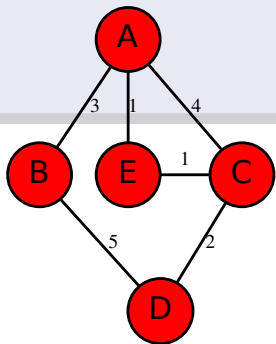
```
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
```



graf g

Afegim un atribut distància a les arestes

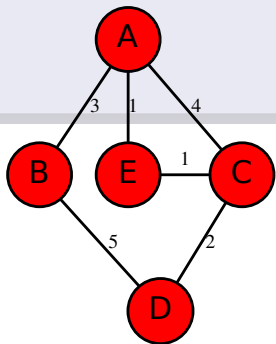
```
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
```



graf g

Afegim un atribut distància a les arestes

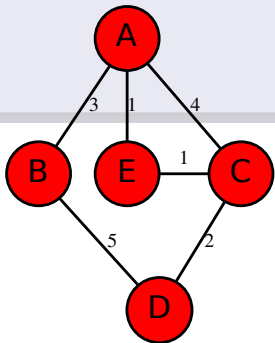
```
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
```



graf g

Afegim un atribut distància a les arestes

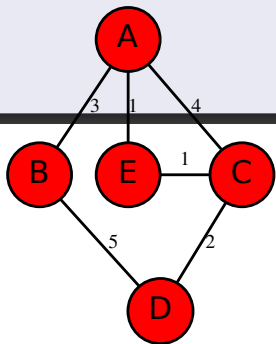
```
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
```



graf g

Afegim un atribut distància a les arestes

```
>>> g['a']['b']['dist'] = 3
>>> g['a']['c']['dist'] = 4
>>> g['a']['e']['dist'] = 1
>>> g['b']['d']['dist'] = 5
>>> g['c']['d']['dist'] = 2
>>> g['c']['e']['dist'] = 1
```



graf g

Consultem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consultem un camí mínim

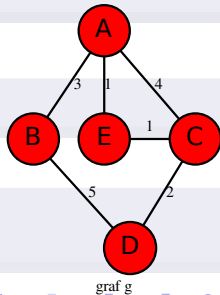
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



Consultem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consultem un camí mínim

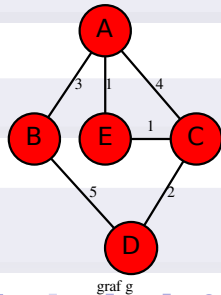
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



Camins mínims amb atributs a valorar

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

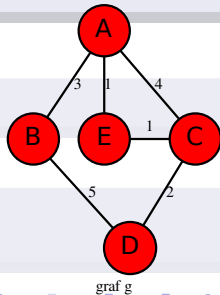
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



Camins mínims amb atributs a valorar

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

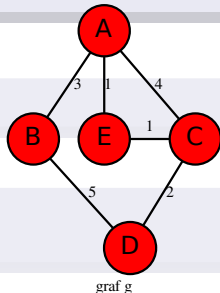
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

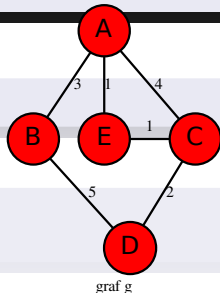
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



graf g

Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

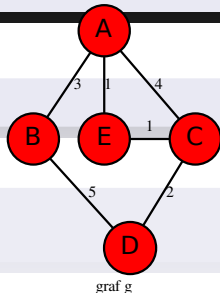
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



graf g

Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

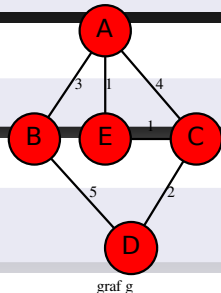
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



graf g

Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

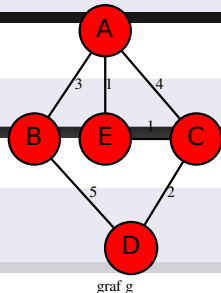
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



graf g

Camins mínims amb atributs a valorar

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre dos vèrtexs

Consulem la longitud del camí mínim segons l'etiqueta 'dist'

```
>>> nx.shortest_path_length(g, 'a', 'd', weight='dist')
4
```

Consulem un camí mínim

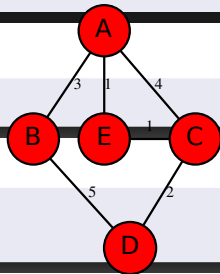
```
>>> nx.shortest_path(g, 'a', 'd', weight='dist')
['a', 'e', 'c', 'd']
```

Afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
```

Hi ha camí entre 'a' i 'f'?

```
>>> nx.has_path(g, 'a', 'f')
False
```



graf g



Components connectats. Graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```



Components connectats. Graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```



Components connectats. Graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```



Components connectats. Graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```

Components connectats. Graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

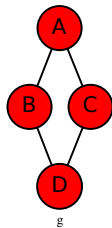
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```



Components connectats. Graf exemple

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

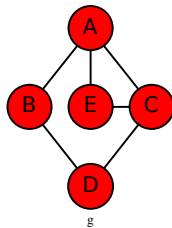
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> nx.add_path(g, ['a', 'b', 'd', 'c', 'a'])
>>> nx.add_path(g, ['a', 'e', 'c'])
```



És connex?

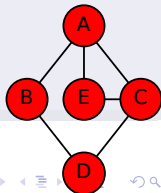
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



És connex?

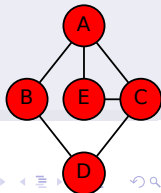
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



És connex?

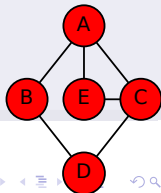
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



És connex?

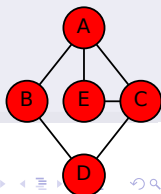
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



És connex?

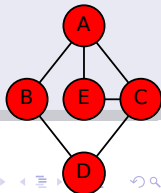
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



És connex?

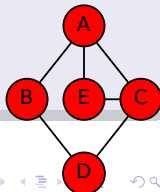
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



És connex?

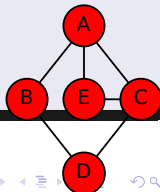
```
>>> nx.is_connected(g)
True
```

Quants components té?

```
>>> nx.number_connected_components(g)
1
```

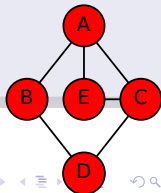
Quins components té?

```
>>> for c in nx.connected_components(g):
...     print(sorted(c))
['a', 'b', 'c', 'd', 'e']
```



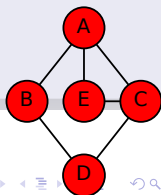
Hi afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
2
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



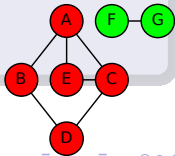
Hi afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
2
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



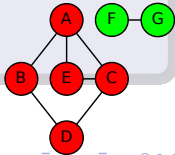
Hi afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
2
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



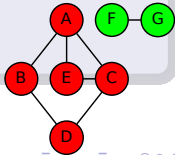
Hi afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
2
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



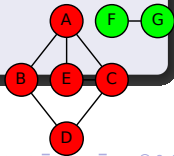
Hi afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
2
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



Hi afegim un altre camí

```
>>> nx.add_path(g, ['f', 'g'])
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
2
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```

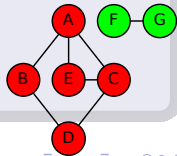


Hi afegim un vèrtex

```

>>> g.add_node('h')
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
3
>>> for c in sorted(nx.connected_components(g), key=len):
...     print(sorted(c))
...
['h']
['f', 'g']
['a', 'b', 'c', 'd', 'e']

```

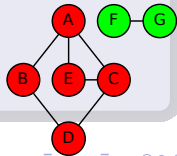


Hi afegim un vèrtex

```

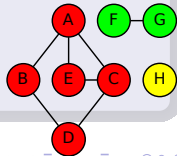
>>> g.add_node('h')
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
3
>>> for c in sorted(nx.connected_components(g), key=len):
...     print(sorted(c))
...
['h']
['f', 'g']
['a', 'b', 'c', 'd', 'e']

```



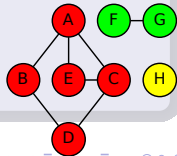
Hi afegim un vèrtex

```
>>> g.add_node('h')
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
3
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['h']
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



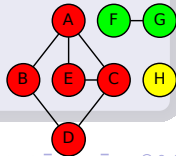
Hi afegim un vèrtex

```
>>> g.add_node('h')
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
3
>>> for c in sorted(nx.connected_components(g), key=len):
...     print(sorted(c))
...
['h']
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



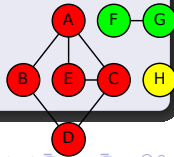
Hi afegim un vèrtex

```
>>> g.add_node('h')
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
3
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['h']
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```



Hi afegim un vèrtex

```
>>> g.add_node('h')
>>> nx.is_connected(g)
False
>>> nx.number_connected_components(g)
3
>>> for c in sorted(nx.connected_components(g), ke
y=len):
...     print(sorted(c))
...
['h']
['f', 'g']
['a', 'b', 'c', 'd', 'e']
```

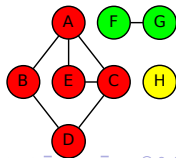


Quin és el component de 'b'?

```
>>> nx.node_connected_component(g, 'b')
{'b', 'a', 'd', 'c', 'e'}
```

I el de 'f'?

```
>>> nx.node_connected_component(g, 'f')
{'g', 'f'}
```

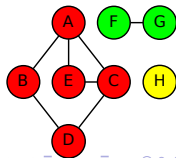


Quin és el component de 'b'?

```
>>> nx.node_connected_component(g, 'b')
{'b', 'a', 'd', 'c', 'e'}
```

I el de 'f'?

```
>>> nx.node_connected_component(g, 'f')
{'g', 'f'}
```

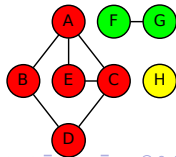


Quin és el component de 'b'?

```
>>> nx.node_connected_component(g, 'b')
{'b', 'a', 'd', 'c', 'e'}
```

I el de 'f'?

```
>>> nx.node_connected_component(g, 'f')
{'g', 'f'}
```

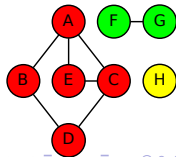


Quin és el component de 'b'?

```
>>> nx.node_connected_component(g, 'b')
{'b', 'a', 'd', 'c', 'e'}
```

I el de 'f'?

```
>>> nx.node_connected_component(g, 'f')
{'g', 'f'}
```

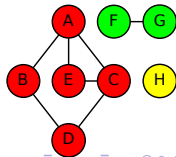


Quin és el component de 'b'?

```
>>> nx.node_connected_component(g, 'b')
{'b', 'a', 'd', 'c', 'e'}
```

I el de 'f'?

```
>>> nx.node_connected_component(g, 'f')
{'g', 'f'}
```





Tipus de grafs

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- **Graf no dirigit (simple):** Les arestes no tenen orientació (parelles sense ordre)

- **Graf dirigit (digraf):** Les arestes tenen orientació (parelles ordenades)

A l'estar les arestes orientades podem parlar de:

- **predecessor directe:** objecte a l'origen de l'aresta
- **successor directe:** objecte al destí de l'aresta
- Si un camí que porta a un vèrtex y , s'inicia en el vèrtex x , tots els vèrtexs successius pels que els passa el camí seran successors de x inclòs y .
- **Multigrafs:** Grafs amb arestes múltiples. Si pot haver dos o més arestes incidents entre dos mateixos vèrtexs, parlem d'**arestes múltiples**. Alguns exclouen de la definició els **cicles** o **bucles** (arestes amb origen i destí al mateix vèrtex) i quan els inclouen parlen de **pseudografs**.



Tipus de grafs

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

- **Graf no dirigit (simple):** Les arestes no tenen orientació (parelles sense ordre)
- **Graf dirigit (digraf):** Les arestes tenen orientació (parelles ordenades)
A l'estar les arestes orientades podem parlar de:
 - **predecessor directe:** objecte a l'origen de l'aresta
 - **successor directe:** objecte al destí de l'aresta
 - Si un camí que porta a un vèrtex y , s'inicia en el vèrtex x , tots els vèrtexs successius pels que els passa el camí seran successors de x inclòs y .
- **Multigrafs:** Grafs amb arestes múltiples. Si pot haver dos o més arestes incidents entre dos mateixos vèrtexs, parlem d'**arestes múltiples**. Alguns exclouen de la definició els **cicles** o **bucles** (arestes amb origen i destí al mateix vèrtex) i quan els inclouen parlen de **pseudografs**.

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

- **Graf no dirigit (simple):** Les arestes no tenen orientació (parelles sense ordre)
- **Graf dirigit (digraf):** Les arestes tenen orientació (parelles ordenades)
A l'estar les arestes orientades podem parlar de:
 - **predecessor directe:** objecte a l'origen de l'aresta
 - **successor directe:** objecte al destí de l'aresta
 - Si un camí que porta a un vèrtex y , s'inicia en el vèrtex x , tots els vèrtexs successius pels que els passa el camí seran successors de x inclòs y .
- **Multigrafs:** Grafs amb arestes múltiples. Si pot haver dos o més arestes incidents entre dos mateixos vèrtexs, parlem d'**arestes múltiples**. Alguns exclouen de la definició els **cicles** o **bucles** (arestes amb origen i destí al mateix vèrtex) i quan els inclouen parlen de **pseudografs**.

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

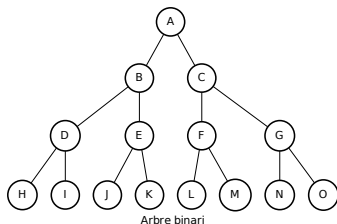
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre dos vèrtexs

Un **arbre** és un graf no dirigit en que cada parell de vèrtexs estan connectats exactament un camí que no té vèrtexs repetits (camí simple).

També es pot definir com un graf connectat (qualsevol parella de vèrtexs els connecta un camí) sense cicles.



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

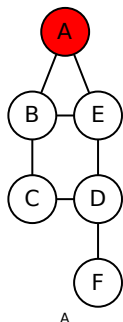
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G,v,marcats)
```

```
def dfs_1(G, v, marcats):
    marcats.add(v)
    visita(v)
    for w in G.neighbors(v):
        if w not in marcats:
            dfs_1(G,w,marcats)
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

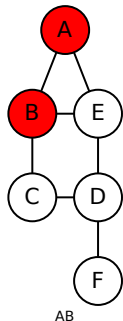
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G,v,marcats)
```

```
def dfs_1(G, v, marcats):
    marcats.add(v)
    visita(v)
    for w in G.neighbors(v):
        if w not in marcats:
            dfs_1(G,w,marcats)
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

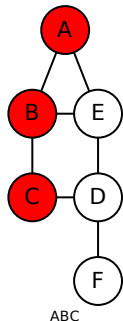
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G,v,marcats)

def dfs_1(G, v, marcats):
    marcats.add(v)
    visita(v)
    for w in G.neighbors(v):
        if w not in marcats:
            dfs_1(G,w,marcats)
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

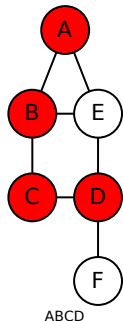
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G,v,marcats)

def dfs_1(G, v, marcats):
    marcats.add(v)
    visita(v)
    for w in G.neighbors(v):
        if w not in marcats:
            dfs_1(G,w,marcats)
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

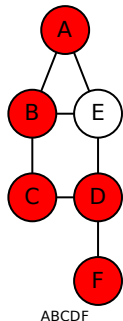
Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G,v,marcats)

def dfs_1(G, v, marcats):
    marcats.add(v)
    visita(v)
    for w in G.neighbors(v):
        if w not in marcats:
            dfs_1(G,w,marcats)
```



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

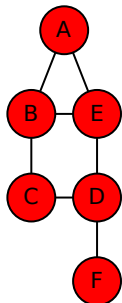
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G,v,marcats)

def dfs_1(G, v, marcats):
    marcats.add(v)
    visita(v)
    for w in G.neighbors(v):
        if w not in marcats:
            dfs_1(G,w,marcats)
```



ABCDFE



Recorregut en primera profunditat (iteratiu)

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

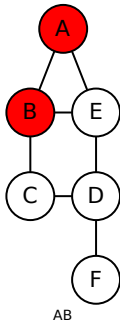
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

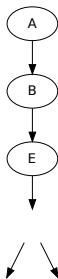
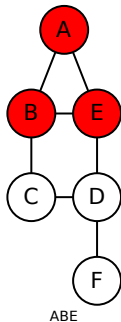
Exercici: Camí menys curt entre

```
def dfs(G,v):  
    pila=[ v ] #Creem pila amb el vertex v  
    marcats=set()  
    while pila!=[]: #no pila buida  
        v=pila.pop() #desempilem  
        if v not in marcats:  
            marcats.add(v)  
            visita(v)  
            for vei in G.neighbors(v):  
                pila.append(vei) #apilem vei
```

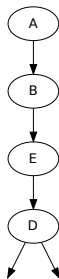
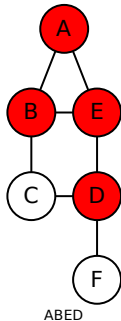

Exemple d'arbre de recorregut en profunditat



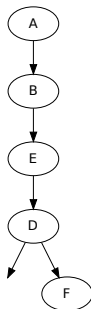
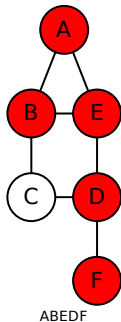
Exemple d'arbre de recorregut en profunditat



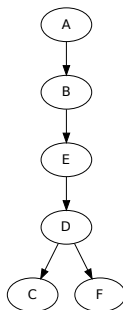
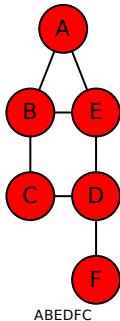
Exemple d'arbre de recorregut en profunditat



Exemple d'arbre de recorregut en profunditat



Exemple d'arbre de recorregut en profunditat





Recorregut primera profunditat com a iterador recursiu

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def dfs(G, v):
    marcats = set()
    return dfs_1(G, v, marcats)

def dfs_1(G, v, marcats):
    marcats.add(v)
    for w in G.successors(v):
        if w not in marcats:
            yield w
            for x in dfs_1(G, w, marcats):
                yield x
```

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

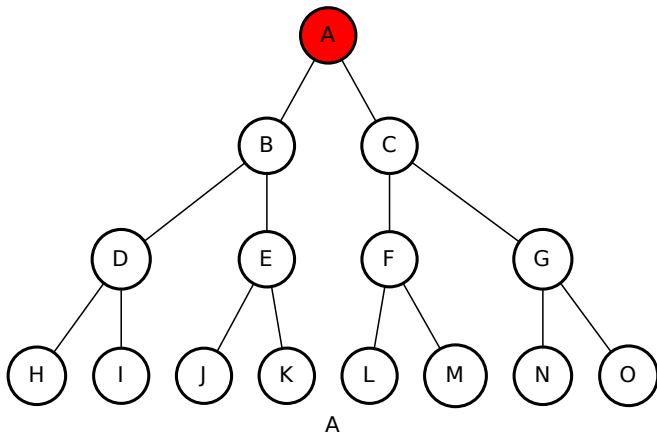
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

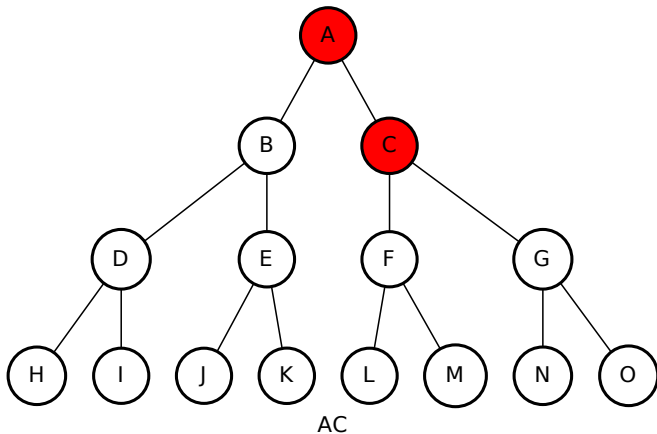
Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Exemple de recorregut sobre un arbre



Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

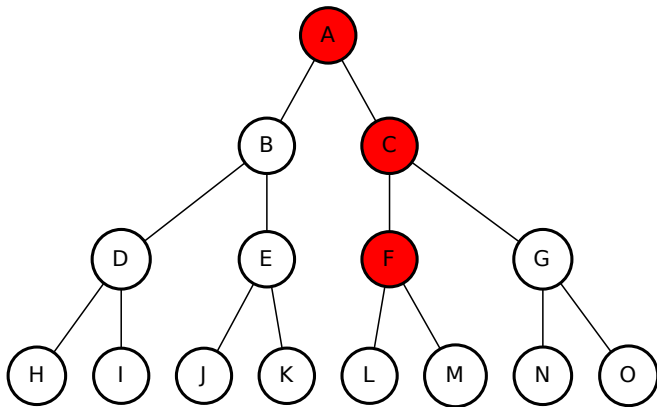
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

Exemple de recorregut sobre un arbre



ACF

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

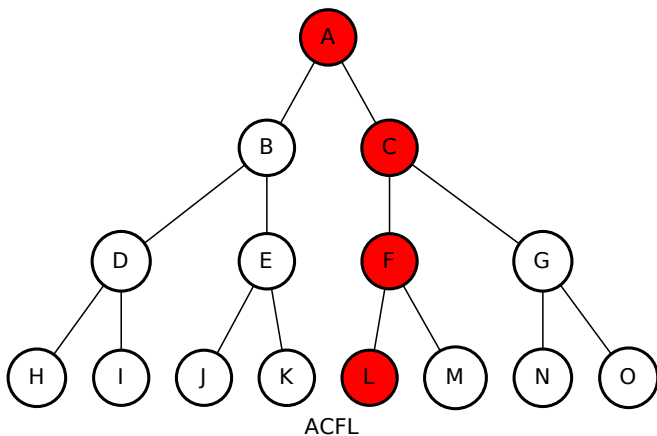
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

Exemple de recorregut sobre un arbre



Conceptes

Introducció al
networkx

Grafs simètric (no
dirigits)

Camins

Components

Tipus de grafs

**Recorregut en
primera
profunditat**

Recorregut en
primera
amplada

Exercici:
Camí més
curt entre dos
vèrtexs

Exercici:
Camí menys
costós entre

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

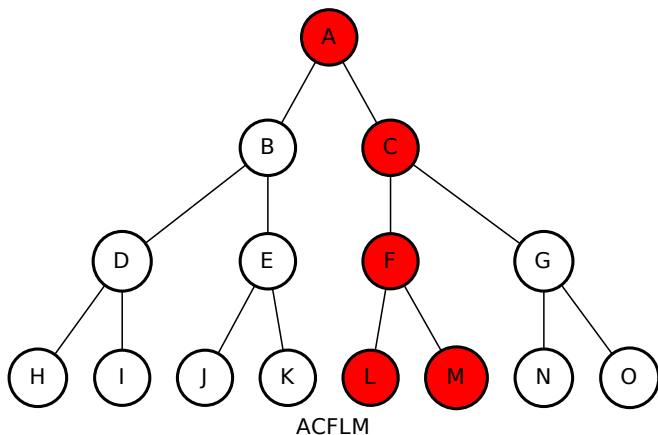
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

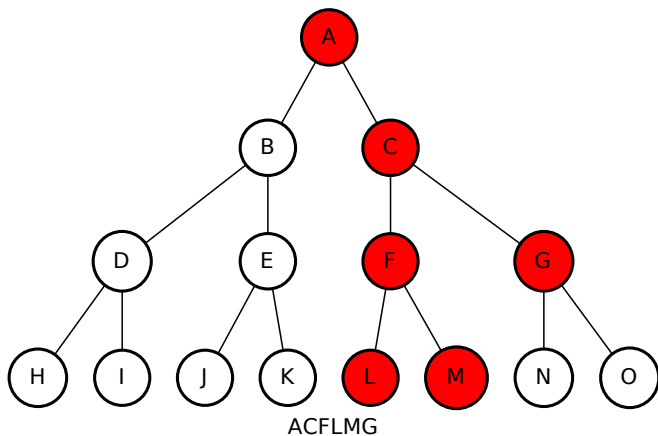
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

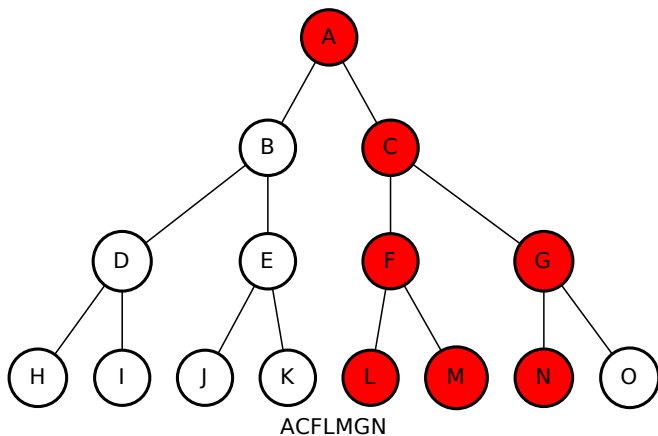
Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Exemple de recorregut sobre un arbre



Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

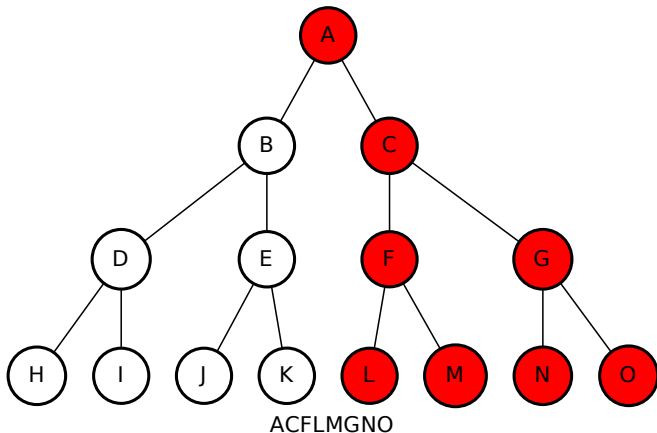
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

Exemple de recorregut sobre un arbre



Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

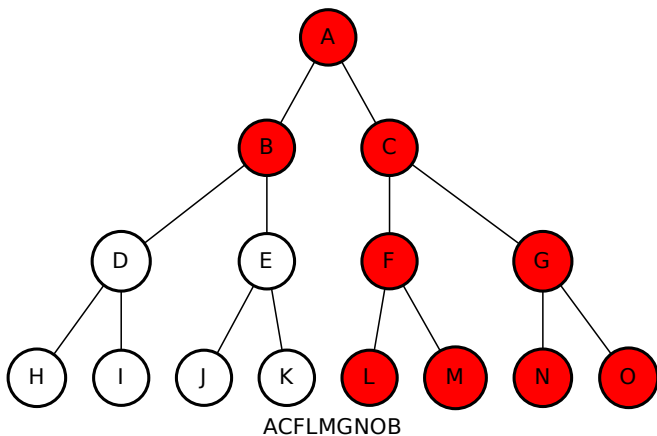
Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

Exemple de recorregut sobre un arbre



Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

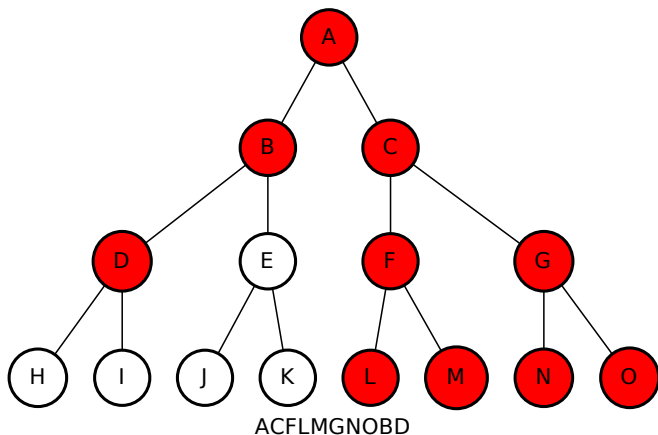
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

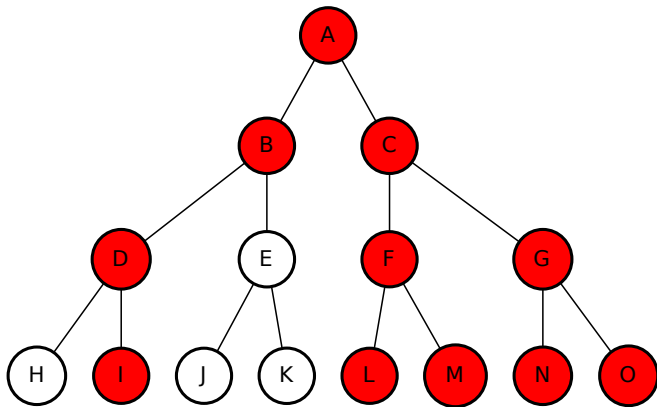
Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Exemple de recorregut sobre un arbre



ACFLMGNOBDI

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

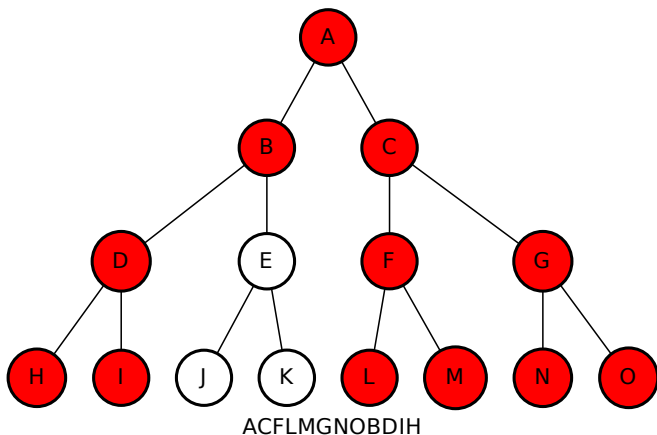
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

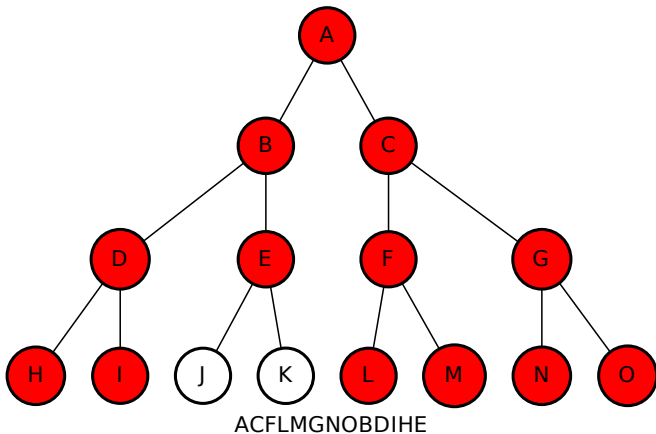
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

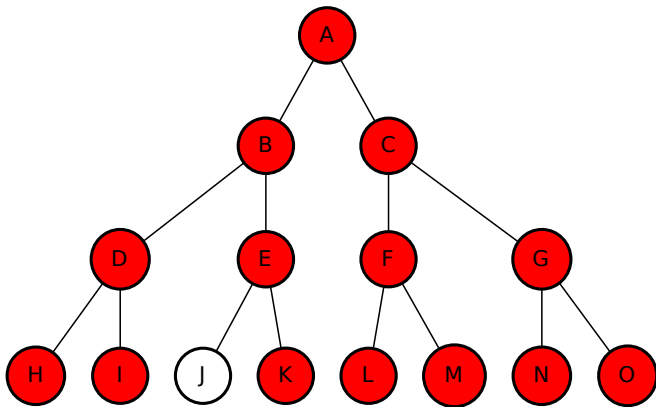
Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Exemple de recorregut sobre un arbre



ACFLMGNNOBBDIHEK

Conceptes

Introducció al
networkx

Grafs simètric (no
dirigits)

Camins

Components

Tipus de grafs

**Recorregut en
primera
profunditat**

Recorregut en
primera
amplada

Exercici:
Camí més
curt entre dos
vèrtexs

Exercici:
Camí menys
costós entre

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

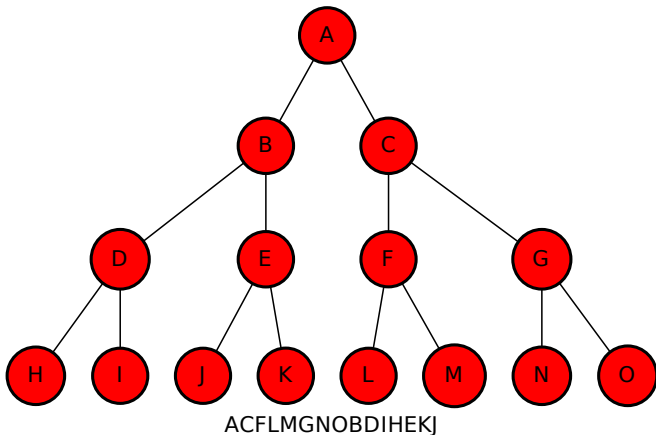
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

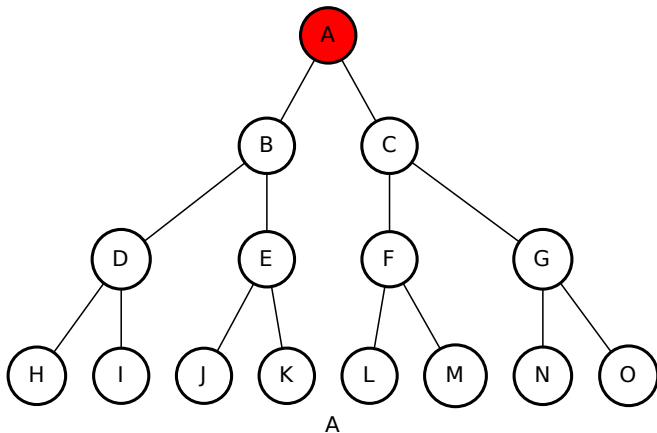
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

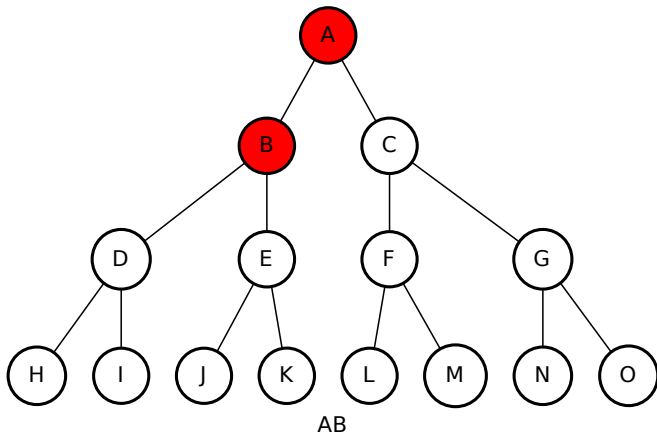
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

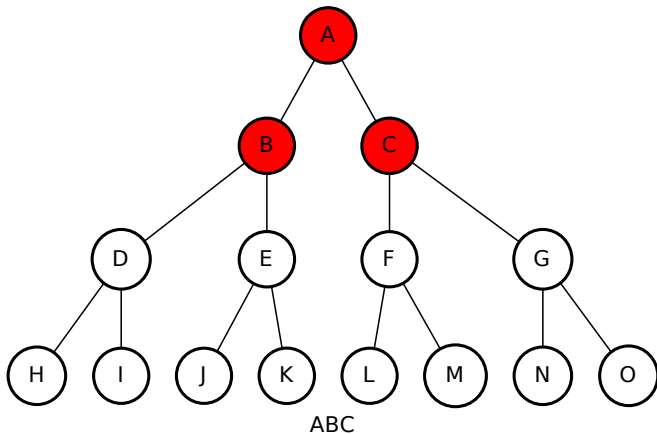
Recorregut en primera profunditat

Recorregut en primera amplada

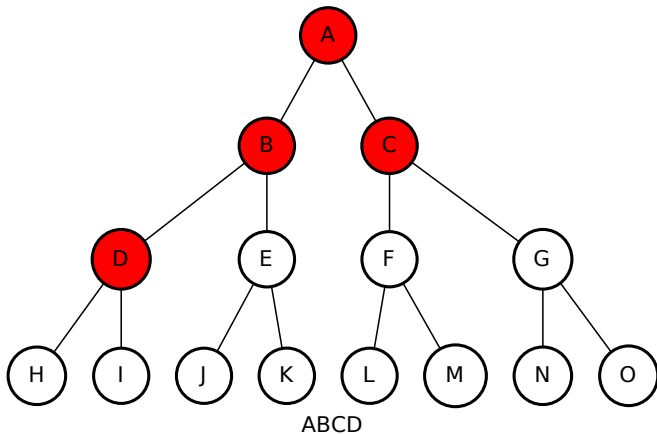
Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

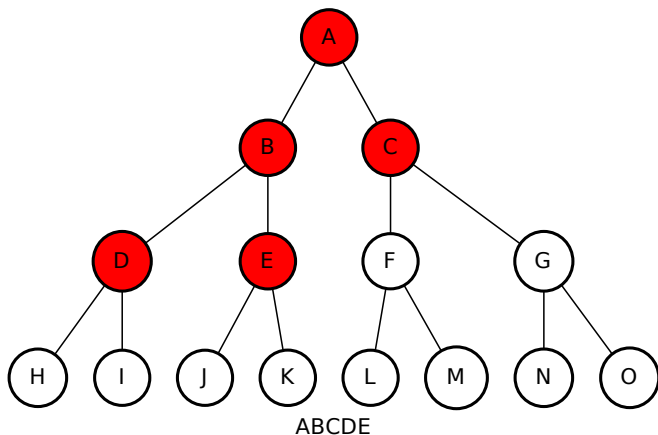
Exemple de recorregut sobre un arbre



Exemple de recorregut sobre un arbre



Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

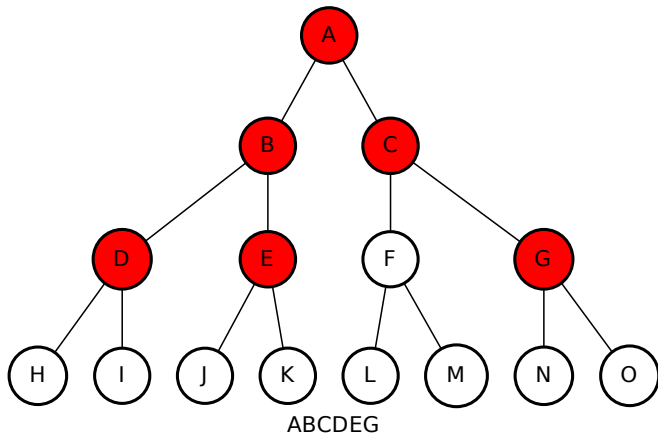
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

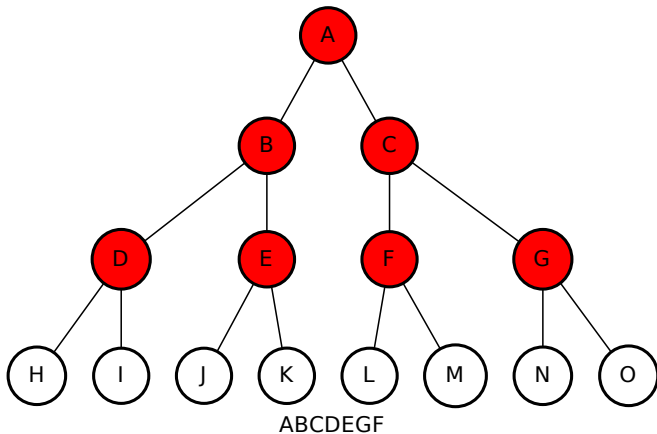
Recorregut en primera profunditat

Recorregut en primera amplada

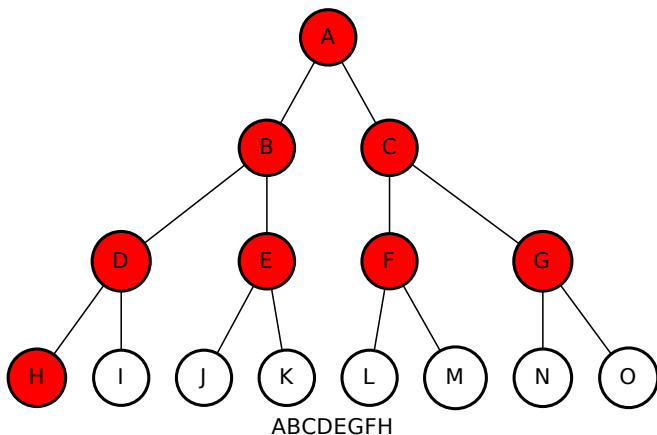
Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

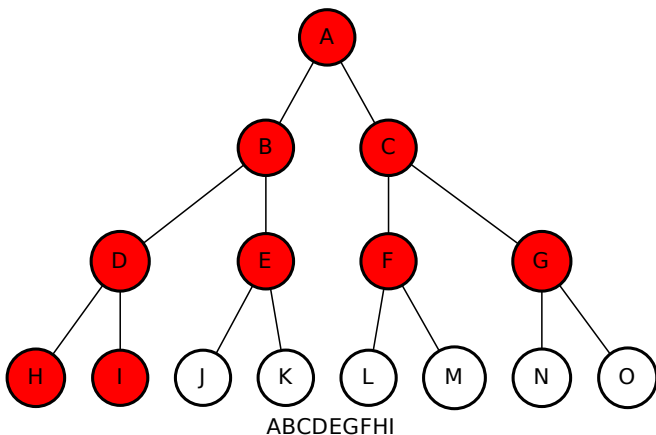
Exemple de recorregut sobre un arbre



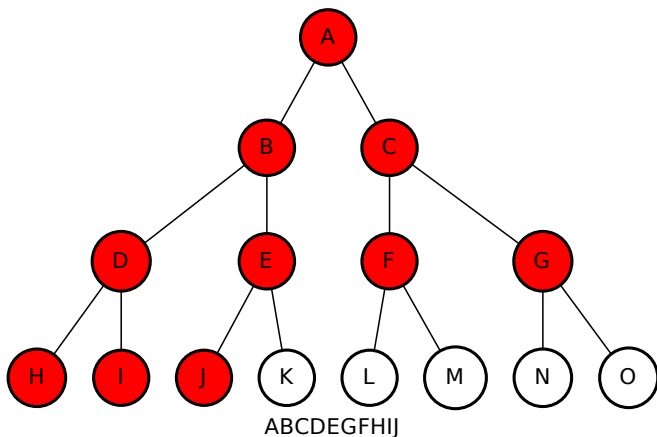
Exemple de recorregut sobre un arbre



Exemple de recorregut sobre un arbre



Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

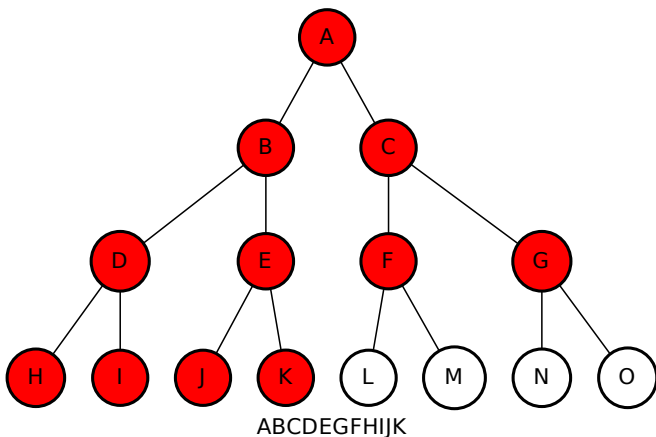
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

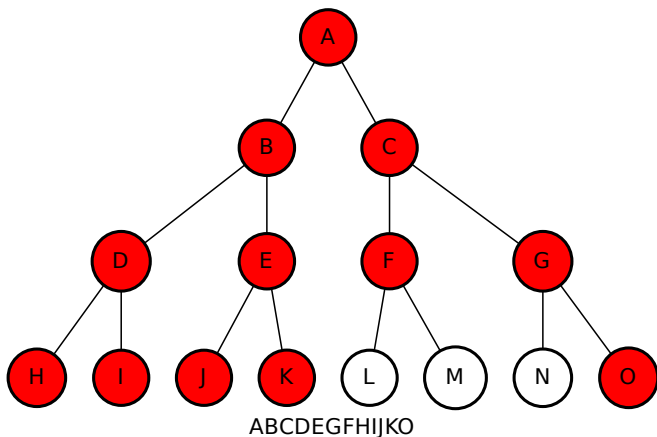
Recorregut en primera profunditat

Recorregut en primera amplada

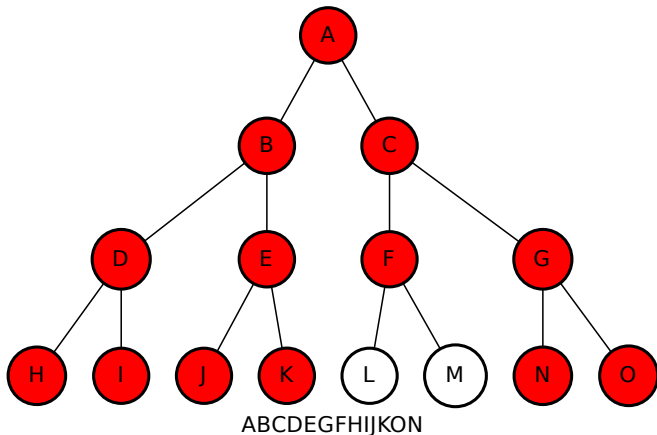
Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

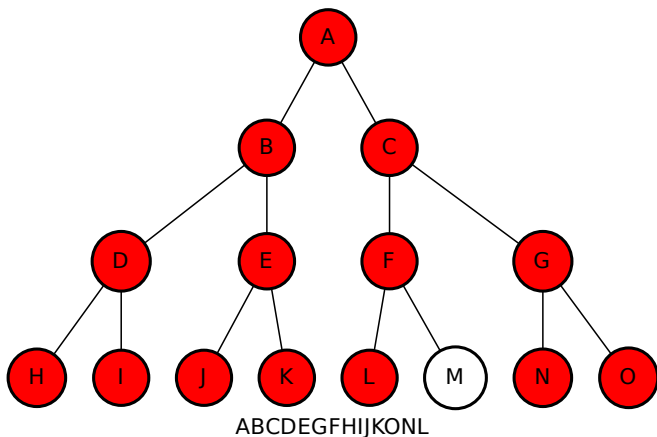
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre



Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

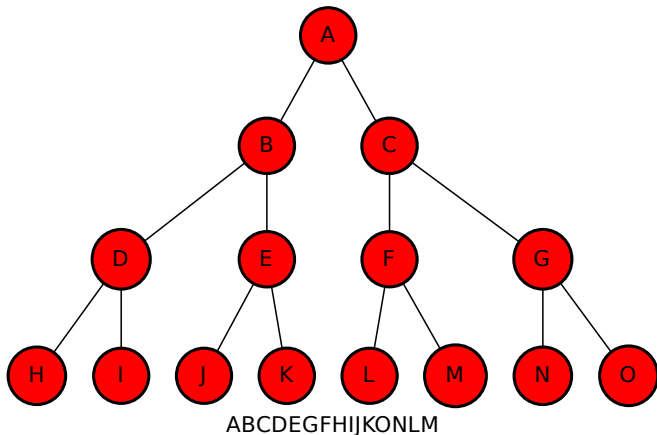
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Exemple de recorregut sobre un arbre





Recorregut en primera amplada

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def bfs(graf, vini):
    cua = [vini] #encua v
    marcats = set(cua)
    while cua != []: #cua no buida
        v = cua.pop(0) # desencua
        yield v
        for w in graf.neighbors(v):
            if w not in marcats:
                marcats.add(w)
                cua.append(w) #encua w
```



Recorregut en primera amplada

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

```
import collections as c

def bfs(graf, v):
    cua = c.dequeue([v]) #encua v
    marcats = set(cua)
    while len(cua) != 0: #cua no buida
        v = cua.popleft() # desencua
        yield v
        for w in graf.neighbors(v):
            if w not in marcats:
                marcats.add(w)
                cua.append(w) #encua w
```



Exercici: Camí més curt entre dos vèrtexs

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

**Exercici:
Camí més curt entre dos vèrtexs**

Exercici:
Camí més curt entre dos vèrtexs

Dissenyar la funció `camiMesCurt(g,ini,fi)` que retorna el camí més curt que hi ha en el graf g que connecta els vèrtexs ini i fi . S'entén com a camí curt aquell conté menys arestes o vèrtexs. El camí retornat estarà representat per una llista amb els vèrtexs del camí seguint l'ordre en que es van trobant per anar des d' ini fins a fi .



Camí més curt entre dos vèrtexs. Plantejament

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Tenim diverses possibilitats de solució:

- Generar tots els possibles camins entre els dos vèrtexs i quedar-se amb aquell que té menys arestes.
- El recorregut BFS tracta sempre els nodes més propers abans de prosseguir amb la resta. Fer una cerca en BFS garanteix que el primer camí trobat entre els dos vèrtexs ja és el camí més curt.



Camí més curt entre dos vèrtexs. Plantejament

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Tenim diverses possibilitats de solució:

- Generar tots els possibles camins entre els dos vèrtexs i quedar-se amb aquell que té menys arestes.
- El recorregut BFS tracta sempre els nodes més propers abans de prosseguir amb la resta. Fer una cerca en BFS garanteix que el primer camí trobat entre els dos vèrtexs ja és el camí més curt.



Camí més curt entre dos vèrtexs. Solució

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys curt entre

```
def camíMesCurt(graf, inici, final):  
    cua=[(inici, [inici])]  
    while cua:  
        (vertex, cami)=cua.pop(0)  
        if vertex==final:  
            return cami  
        for v in graf.neighbors(vertex):  
            if v not in cami:  
                cua.append((v, cami+[v]))
```



Camí més curt entre dos vèrtexs. Solució

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

```
from collections import deque
def camíMesCurt(graf, inici, final):
    cua=deque([(inici, [inici])])
    while cua:
        (vertex, cami)=cua.popleft()
        if vertex==final:
            return cami
        for v in graf.neighbors(vertex):
            if v not in cami:
                cua.append((v, cami+[v]))
```

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

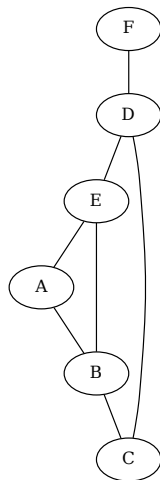
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

```
>>> from camiMesCurt import
>>> import networkx as nx
>>> g=nx.Graph()
>>> nx.add_path(g,['F','D',
... 'A','B','C','D'])
>>> g.add_edge('B','E')
>>> camiMesCurt(g,'A','F')
['A', 'E', 'D', 'F']
>>> camiMesCurt(g,'A','D')
['A', 'E', 'D']
>>> camiMesCurt(g,'A','A')
['A']
>>> camiMesCurt(g,'B','F')
['B', 'C', 'D', 'F']
```





Camí més curt entre dos vèrtexs. Solució

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys curt entre

De fet, *networkx* té molts algorismes típics pel tractament de grafs. I podem solucionar el problema cridant directament *shortest_path*:

```
import networkx as nx

def camíMesCurt(graf, inici, final):
    return shortest_path(graf,
                          source=inici,
                          target=final)
```

o

```
import networkx as nx

def camíMesCurt(graf, inici, final):
    return shortest_path(graf, inici, final)
```



Exercici: Camí menys costós entre dos vèrtexs

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

Dissenyar la funció `camiMesCurt(g,ini,fi)` que retorna el camí més curt que hi ha en el graf g que connecta els vèrtexs ini i fi . Cada aresta del graf té l'atribut `cost` amb el valor de quant costa travessar-la. S'entén com a camí curt aquell conté el cost mínim per arribar a fi des d' ini . El camí retornat estarà representat per una llista amb els vèrtexs del camí seguint l'ordre en que es van trobant per anar des d' ini fins a fi .



Exercici: Camí menys costos entre dos vèrtexs

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

Podem solucionar el problema cridant directament *shortest_path*:

```
import networkx as nx

def camíMesCurt(graf, inici, final):
    return shortest_path(graf,
                        source=inici,
                        target=final
                        weight='cost')
```

o

```
import networkx as nx

def camíMesCurt(graf, inici, final):
    return shortest_path(graf, inici, final, 'cost')
```

En *weight* posem el nom de l'atribut de l'aresta que cal avaluar.



Camí menys costós entre dos vèrtexs. Solució

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def caminsDFS(graf, inici, final):
    cami=[inici]
    return camins(graf, inici, final, cami, 0)

def camins(g, ini, fi, cami, cost):
    if ini == fi:
        yield cami, cost
    for v in g.neighbors(ini):
        if v not in cami:
            for via, c in camins(g, v, fi,
                                cami+[v],
                                cost+g[ini][v]['cost']):
                yield via, c
```



Camí menys costós entre dos vèrtexs. Solució

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
def camiMesCurt(graf, inici, final):  
    caminsCost= caminsDFS(graf, inici, final)  
    camiCurt= next(caminsCost)  
    for camiCost in caminsCost:  
        if camiCost[1]<camiCurt[1]:  
            camiCurt = camiCost  
    return camiCurt[0]
```



Camí menys costós entre dos vèrtexs. Proves

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no
dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici: Camí més curt entre dos vèrtexs

Exercici: Camí menys costós entre

```
>>> from camiMesCurt2 import camiMesCurt
>>> import networkx as nx
>>> g=nx.Graph()
>>> costos=[1,2,3,2,1,2]
>>> cami= ['F', 'D', 'E', 'A', 'B', 'C', 'D']
>>> nx.add_path(g, cami)
>>> icami=iter(cami)
>>> origen= next(icami)
>>> for desti, cost in zip(icami,costos):
...     g[origen][desti]['cost']=cost
...     origen=desti
...
>>> g.add_edge('B','E',cost=3)
```

Camí menys costós entre dos vèrtexs. Proves

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

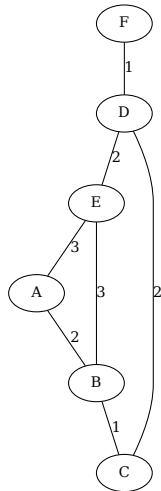
Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

```
>>> camiMesCurt(g, 'A', 'F')
['A', 'E', 'D', 'F']
>>> camiMesCurt(g, 'A', 'D')
['A', 'E', 'D']
>>> camiMesCurt(g, 'A', 'A')
['A']
>>> camiMesCurt(g, 'B', 'F')
['B', 'C', 'D', 'F']
```



Camí menys costós entre dos vèrtexs. Proves

Grafs

Conceptes

Introducció al networkx

Grafs simètric (no dirigits)

Camins

Components

Tipus de grafs

Recorregut en primera profunditat

Recorregut en primera amplada

Exercici:
Camí més curt entre dos vèrtexs

Exercici:
Camí menys costós entre

```
>>> camiMesCurt(g, 'A', 'F')
['A', 'B', 'C', 'D', 'F']
>>> camiMesCurt(g, 'A', 'D')
['A', 'B', 'C', 'D']
>>> camiMesCurt(g, 'A', 'A')
['A']
>>> camiMesCurt(g, 'B', 'F')
['B', 'C', 'D', 'F']
```

