UPC

PROGRAMACIÓ
Grau en Estadística UB-UPC, febrer 2021
Prof Robert Joan-Arinyo

Matrius

Exercici 1

Dissenya una funció tal que donada una matriu d'enters, `m`, determini els índexs de la component amb valor màxim.

Exercici 2

Escriu una funció `R` tal que donada una matriu de cadenes de caràcters, `m`, determini i retorni els índexs de la primera component que contingui la cadena `'abc'`. Si no existeix aital cadena, la funció retornarà l'enter -1.

Exercici 3

Implementa la funció `esEnMatriu(lletra, matriu)` definida amb els següents examples:

```
m <- [['a', 'b'], ['c', 'd']]
c <- 'a'
esEnMatriu(m, c) retorna  True

m <- [['a', 'b'], ['c', 'd']]
c <- 'b'
esEnMatriu(m, c) retorna True

m <- [['a', 'b'], ['c', 'd']]
c <- 'c'
esEnMatriu(m, c) retorna True

m <- [['a', 'b'], ['c', 'd']]
c <- 'd'
esEnMatriu(m, c) tretorna  True

m <- [['a', 'b'], ['c', 'd']]
```

```
    c <- 'x'
    esEnMatriu(m, c) retorna  False
  """
```

Exercici 4

Dissenya la funció `sumaMatrius(m, n)` tal que donades dues matrius de valors enters, `m` i `n`, calcula i retorna la suma d'aquestes matrius.

Exercici 5

Dissenya la funció `producteMatrius(m, n)` tal que donades dues matrius de valors enters, `m` i `n`, calcula i retorna el producte d'aquestes matrius.

Exercici 6

Hom diu que una matriu és *triangular superior* si tots els elements situats per sota de la diagonal principal tenen valor nul. Per exemple

$$m = \begin{pmatrix} 7 & 12 & -3 & 5 \\ 0 & 4 & 2 & 0 \\ 0 & 0 & -2 & 81 \\ 0 & 0 & 0 & 6 \end{pmatrix}$$

Dissenya la funció `esTriangular(m)` tal que, donada la matriu `m`, retorna `True` si la matriu és triangular superior i `False` en cas contrari.

Exercici 7

Hom diu que una matriu és *diagonal* si tots els elements que no pertanyen a la diagonal principal tene valor nul.

$$m = \begin{pmatrix} 7 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 6 \end{pmatrix}$$

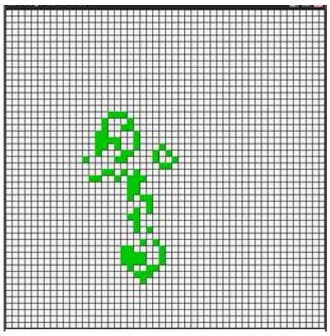
Dissenya la funció `esDiagonal(m)` tal que, donada la matriu `m`, retorna `True` si la matriu és diagonal i `False` en cas contrari.

Exercici 8

Cap al 1970 el matemàtic britànic John Conway definí un conjunt simple de regles per tal de simular el comportament vegetatiu de plantes i mcroorganismes. Les regles d'evolució són

1. Sobrepoblació: si una planta està envoltada per més de tres plantes, mor.
2. Supervivència: si una planta està envoltada per dues o tres plantes, la planta sobreviu.
3. Reproducció: si un espai on no hi ha planta està envoltat exactament per tres plantes, neix una nova planta.
4. Despoblació: si una planta està envoltada per menys de dues plantes mor per solitud.

Si suposem que la distribució de les unitats biològiques està en un espai bidimensional, hom pot assumir que estan localitzades en una graella uniforme i l'estat de cada planta s'emmagatzema en una component d'una matriu de caràcters. Per exemple, la presència de planta es representa amb el caràcter '*' i l'absència de planta amb el caràcter blanc.



Font: Wikipedia

Si `m` és la matriu on es desenvolupa l'evolució, el veinatge general de la planta situada a la component `m[i][j]` està definit per vuit componets amb índexs: (i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), (i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1).

És evident que aquest veinat no és correcte quan la planta està a una componet de la matriu on algun índex és ú o igual a la darrera component. Per tal que aquestes regles siguin sempre vàlides, la matriu del joc s'amplia amb un marc de files i columnes que formen una frontera exterior i que emmagatzemen una informació especial que indica que és la frontera. Hom diu que ha orlat la matriu donada.

Programa el joc.

Exercici 9

Un antic problema de matemàtica combinatòria es preguntava: és possible posicionar vuit reines en un taulell d'escacs de manera que cap en mati cap?

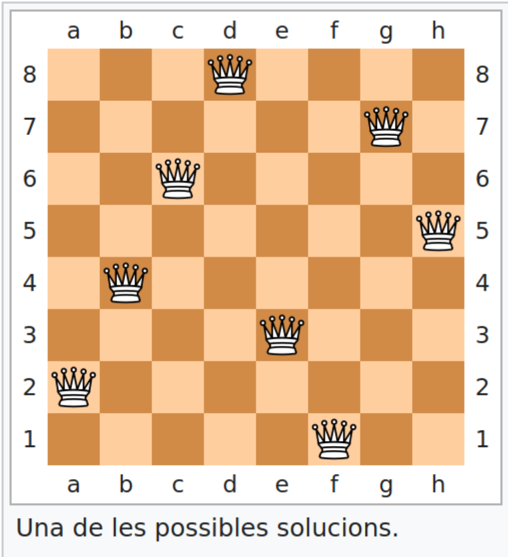


Figura 1: Font: Viquipèdia

Se sap que hi ha solució sempre que les dimensions del taulell siguin $n = 1$ o $n \geq 4$. El cas usual del taulell de 8×8 té 92 solucions diferents. La Figura 1 en mostra una. Dissenya un programa que trobi una solució a partir de posicionar la primera reina en la primera columna de la primera fila del taulell.

Exercici 10

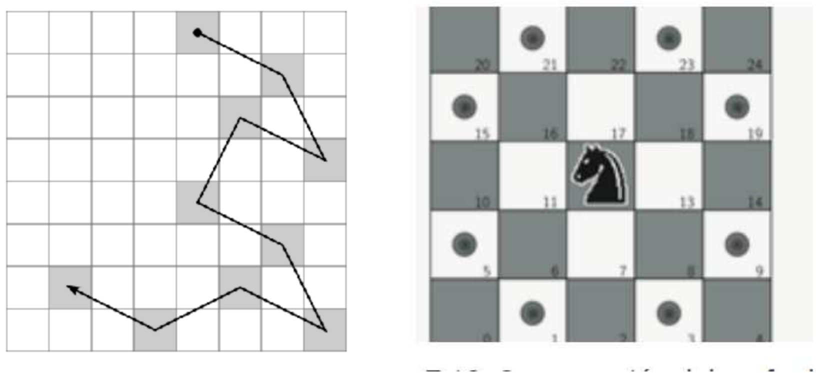
Les trajectòries que el moviment del cavall d'escacs pot definir sobre el taulell de joc genera tot un seguit de problemes en els que la *matemàtica discreta* està interessada i que es poden denominar com *trajectòries del cavall*, de l'anglès *Knight's tour*. Un dels més populars d'aquests problems s'enuncia com segueix:

> Éxisteix alguna trajectòria tal que el cavall visiti cada casella del taulell un i només que un cop?

La resposta és positiva sempre que s'apliqui una estratègia adient. Donada una posició del cavall pel parell d'índexs (i, j), el nombre màxim de moviments és de vuit, com il.lustra la Figura 2b. Aquests moviments estan definits pels índexs: (i-2, j+1), (i-2, j-1), (i-1, j-2), (i+1, j-2), (i+2, j-1), (i+2, j+1), (i+1, j+2), (i-1, j+2)

Aleshores, per a qualsevol posició de sortida del cavall, l'estratègia següent assegura lèxit:

> Sgui (i, j) la parella d'índexs que defineixen la posisició actual del cavall en el taulell d'escacs. Designem com a moviment candidat cada possible moviment del cavall des de la posició actual. Aleshores, hom determina el nombre possible de



(a) (b)

Figura 2: Font: Viquipèdia. (a) Una trajectòria arbitrària. (b) Els vuit possibles moviments.

moviments des de cada moviment candidat. Finalmenr es tria com a moviment efectiu aquell que mou el cavall a la posició candidata des de la qual el nombre de moviments en el següent pas és màxim.

Programa el joc amb les següents codificacions:

1. Posició no visitada: `taulell[i][j] = 0`.
2. Posició visitada: `taulell[i][j] = i`, on `i` és l'ordinal del moviment. 1, 2, 3, ...,64.
3. Posició no visitable: `taulell[i][j] = -1`.

Per a que els moviments del cavall indicats siguin vàlids independentment de la posició del cavall, cal orlar la matriu de joc amb quatre files i quate columnes, dues per cada costat del taulell. Marqueu-les com a components no visitables.