

Exercicis de Programació 2:

Estructures lineals II: Llistes

Ricard Gavaldà

27 de febrer de 2019

Preguntes prèvies de comprensió:

- Mireu-vos i assegureu-vos que enteneu l'especificació de la classe Llista. És als apunts i a la meua pàgina de pro2, per exemple.

Exercici 1. Feu una acció que, donada una llista de strings i dos strings `a` i `b`, substitueixi tota aparició d'`a` per una de `b`.

Ja sabeu la resposta a la pregunta “cal especificar-la?”, oi?

Exercici 2. Feu una funció que donada una llista d'ints digui si és estrictament creixent.

Exercici 3. Feu dues versions d'una acció que revessi una llista (és a dir, que per exemple transformi `[a,b,c,d]` en `[d,c,b,a]`). Una ha de ser recursiva i l'altra iterativa. El resultat ha de quedar en la mateixa llista.

Exercici 4. Feu una acció que donada una llista de strings i dos strings `ini` i `fi` (que ens prometen que són diferents), busqui totes les aparicions d'`ini` seguides d'alguna de `fi` i elimini de la llista tot el que hi ha entre elles. Per exemple, amb la llista `[a,b,c,b,a,d,f,c,b,b,a,f,d,c,a,b]`, `ini = "c"` i `fi = "d"`, la llista s'ha de transformar en `[a,b,c,d,f,c,d,c,a,b]`.

Exercici 5. Feu una funció que digui si una llista es capicua sense usar la funció `.size()`. Cada element s'ha de consultar un cop com a molt. Recordeu

que no es poden comparar iteradors amb $<$ (nomes amb $==$ i $!=$), ni es pot fer aritmètica amb iteradors (per exemple `it = (l.begin() + l.end())/2`).

Exercici 6. Considereu el següent codi per ordenar un vector de doubles pel metode de selecció:

```
for (int i = 0; i < v.size()-1; ++i) {
    int min = i;
    for (int j = i+1; j < v.size(); ++j)
        if (v[j] < v[min]) min = j;
    swap(v[i],v[min]);
}
```

Recordeu que el metode de selecció tarda temps proporcional al quadrat de `v.size()`.

Traduïm aquest codi a llistes de la següent manera:

```
for (int i = 0; i < l.size()-1; ++i) {
    int min = i;
    for (int j = i+1; j < l.size(); ++j)
        if (l.get(j) < l.get(min)) min = j;
    swap(l,i,min);
}
```

Estem usant l'operació `l.get(index)` que apareix a les transparències i una operació `swap(llista, index1, index2)` que no hem donat a teoria pero que fa el que podeu imaginar, on `index`, `index1` i `index2` són enters entre 0 i `l.size()`.

1. Implementeu l'operació `swap(l, index1, index2)`.
2. Què hi ha de dolent en aquesta implementació?
3. Doneu un codi alternatiu sense aquest problema.

Exercici 7. Volem fer un programa que llegeix doubles per teclat i els deixa en una llista en l'ordre contrari al de lectura.

Fem aquesta solució, correcta però absurda: Primer, fem una operació

```
list<double> concatena(list<double> l1, list<double> l2)
```

que retorna una llista amb la concatenació de l1 i l2. Després, la fem servir d'aquesta manera:

```
double x;
list<double> l;

while (cin >> x) {
    list<double> l2;
    l2.insert(l2.begin(),x);
    l = concatena(l2,l);
}
```

1. Digueu totes les fonts d'ineficiència d'aquesta solució. En particular, digueu quantes còpies de llistes es fan, i digueu quin temps tarda en funció del nombre d'elements llegits.
2. Doneu una solució bona per al mateix problema.