

Listes

Rappels :

Construire une liste, avec L une liste : (cons e L)

Premier élément d'une liste L : (car L)

Reste de la liste L (c'est une liste également) : (cdr L)

Liste vide : '()

Prédicat liste vide : (null? L)

Le dernier « cdr » d'une liste est la liste vide '().

Exercices

- 1) La fonction (iota a b) prenant deux entiers a et b et retournant la liste des entiers [a,b] :
`(iota 5 10) -> (5 6 7 8 9 10)`
- 2) La fonction (\$append L1 L2) qui concatène deux listes L1 et L2 :
`($append '(do re mi) '(fa sol si la re)) -> (do re mi fa sol si la re)`
Expliquez pourquoi la complexité de \$append dépend de la longueur de L1 ?
- 3) La fonction (\$reverse L) qui retourne une copie inversée de la liste L :
`($reverse '(do re (mi fa) sol)) -> (sol (mi fa) re do)`
- 4) La fonction (\$list-ref L k) retournant le k ième élément de la liste L sachant que le premier élément a pour indice 0.
`($list-ref (iota 5 10) 3) -> 8`
- 5) La fonction (k-premiers L k) qui retourne la liste composées des k premiers éléments de L

Tri de liste

1) tri par insertion.

Exemple : `(tri-ins '(5 3 7 2 1 3 4)) -> (1 2 3 3 4 5 7)`

Nous voulons trier une liste L. Supposons par hypothèse de récurrence le problème résolu pour (cdr L) :

`(3 7 2 1 3 4) -> (1 2 3 3 4 7)`

Que reste-t-il à faire ? Facile : il suffit d'insérer le premier élément (car L) dans la liste triée (1 2 3 3 4 7).

A vous de jouer (tri-ins L) !

2) tri fusion

Principe : Pour trier une liste L, on distribue ses éléments dans deux listes L1 et L2 de même longueur ou presque. On trie [par hypothèse de récurrence] L1 et L2. On fusionne les listes triées obtenues.

Exemple, soit L = (7 3 5 9 2 1 8 0 4)

Alors L1 = (4 8 2 5 7) et L2 = (0 1 9 3)

Après tri par hypothèse de récurrence,

LT1 = (2 4 5 7 8) et LT2 = (0 1 3 9)

On peut enfin fusionner LT1 et LT2 : LT = (0 1 2 3 4 5 7 8 9)

- a. Programmez la fonction (scission L) retournant les deux résultats (L1 L2). Les deux listes auront le même nombre d'éléments ou bien l'une des deux en aura un de plus :

`(scission '(7 3 5 9 2 1 8 0 4)) -> ((4 8 2 5 7) (0 1 9 3))`

- b. Programmez la fonction (fusion LT1 LT2) prenant deux listes triées LT1 et LT2 et retournant la liste triée obtenue en fusionnant les deux :

`(fusion '(2 4 5 7 8) '(0 1 3 9)) -> (0 1 2 3 4 5 7 8 9)`

- c. Programmez la fonction (tri-fusion L).