DEUG 1^{ere} année
MIAS 1 et MASS 1 – Programmation fonctionnelle

TD n° 7

Thème : Évaluation de la complexité en temps et en espace

Dans les exercices qui suivent, on précisera en fonction de quelle mesure des données on évalue les complexités, on explicitera les équations de récurrence permettant de calculer ces complexités et après les avoir résolues on donnera leur ordre de grandeur en utilisant la notation O.

Exercice 1 : Évaluer la complexité en temps et en espace de la fonction suivante :

```
(define ajoutefin
  (lambda (o l)
    (if (null? l)
        (cons o l)
        (cons (car l) (ajoutefin o (cdr l))))))
```

Exercice 2 : Évaluer la complexité en temps des fonctions suivantes :

```
(define sauf
  (lambda (o l)
    (cond ((null? l) '())
          ((equal? o (car l)) (sauf o (cdr l)))
          (else (cons (car l) (sauf o (cdr l)))))))
```

```
(define enleve
  (lambda (l1 l2)
    (if (null? l1)
        l2
        (enleve (cdr l1) (sauf (car l1) l2)))))
```

Exercice 3 :
1. La fonction **insere** dont on donne la spécification prend comme arguments un entier e et une liste l d'entiers triés par ordre croissant et insère e à sa place dans l. Écrire en Sheme cette fonction et évaluer sa complexité en temps.

```
; insere : Entier, Liste --> Liste
;           e   ,   l   |-> (e)                 si l est la liste vide
;                           (e . l)              si l=(x . l1) et e <= x
;                           (x . insere(e, l1)) si l=(x . l1) et e > x
```

2. Écrire la fonction **tri**, dont la spécification suit et qui prend en argument une liste d'entiers et renvoie la liste de ces entiers triés en ordre croissant. Évaluer la complexité en temps de la fonction **tri**?

```
; tri : Liste --> Liste
;         l   |-> () si l est la liste vide
;                 (insere x (tri l1)) si l=(x . l1)
```

3. Si la liste donnée en entrée à la fonction **tri** est triée en ordre croissant, que devient la complexité en temps de la fonction **tri**? (On appelle cette complexité la complexité dans le meilleur des cas).