

# TD 1 d'Algorithmique

Lundi 27 septembre 2004

## 1 Listes en C

On définit une structure de liste d'entiers en C de la façon suivante:

```
struct noeud {
    int elem;
    struct noeud *suiv;
};

typedef struct noeud *liste;
```

### Fonctions de base: constructeurs, accesseurs

Écrivez les fonctions suivantes:

- `new_list` (créer une nouvelle liste),
- `is_empty` (tester si une liste est vide),
- `cons` (ajouter un élément en tête d'une liste),
- `first` (récupérer l'élément de tête d'une liste),
- `tail` (récupérer la liste privée de son élément de tête),
- `concat` (concaténer deux listes).

### Fonctions évoluées

- Quel serait le type d'une liste de listes d'entiers? Écrivez une fonction `merge` qui met bout à bout les éléments d'une liste de listes.
- Au niveau des performances, que pensez-vous de notre type de liste? Une suggestion d'amélioration?
- Écrivez la fonction `map` qui prend une liste d'entiers, une fonction `f` des entiers vers les entiers et renvoie la liste des images par `f` des entrées.
- Écrivez la fonction `miroir` qui renverse une liste.
- Écrivez la fonction `reduce` qui applique une fonction `f` à deux arguments de façon cumulée sur une liste. Par exemple, si `f` est l'addition `reduce` devra faire la somme de toute la liste.

### Tri insertion

- Écrire une fonction `insertion` qui insère un élément à la bonne place dans une liste triée. Quelle est la complexité de l'opération d'insertion?

- En déduire une fonction de tri, `tri_insertion`. Quelle est sa complexité? Quel est le pire cas?

### Tri fusion

- Écrivez la fonction `coupe` qui coupe une liste en deux listes de taille égale (à un près quand la liste d'entrée contient un nombre impair d'éléments).
- Écrivez la fonction `fusion` qui prend deux listes triées et renvoie la liste triée contenant les éléments des deux listes d'entiers.
- En déduire une fonction de tri. Quelle est sa complexité? Y a-t-il un pire cas?

## 2 Tableaux en C

### Quelques algos de base

- Écrivez un algorithme pour retrouver l'indice d'un élément d'un tableau trié.
- Écrivez un algorithme pour trouver l'indice de l'élément le plus grand d'un tableau. Et si on veut le deuxième plus grand aussi?
- On suppose que les éléments  $i$  à  $j$  du tableau sont triés, de même que les éléments  $j + 1$  à  $k$ . Écrivez une procédure de fusion pour que les éléments  $i$  à  $k$  du tableau soient triés. On pourra écrire la sortie dans un tableau annexe.
- En déduire une procédure qui trie un tableau d'entiers donné en entrée.

### Arithmétique

On stocke des polynômes de  $\mathbb{Z}[X]$  dans un tableau d'entiers.

- Écrivez une fonction réalisant la somme de deux polynômes.
- Écrivez une fonction pour multiplier deux polynômes (de façon naïve). Quel est la complexité de cette fonction?
- Multiplication de Karatsuba. On rappelle le principe de la multiplication Karatsuba de deux polynômes  $P(X)$  et  $Q(X)$  de degré  $2^n - 1$  chacun:

$$P(X) = P_0(X) + X^{2^{n-1}} P_1(X)$$

$$Q(X) = Q_0(X) + X^{2^{n-1}} Q_1(X)$$

$$P \cdot Q = P_0 \cdot Q_0 + X^{2^{n-1}} (P_0 \cdot Q_1 + P_1 \cdot Q_0) + X^{2^n} P_1 \cdot Q_1$$

semble coûter 4 multiplications de degré  $2^{n-1} - 1$ , mais en calculant:

$$R = P_0 + P_1$$

$$S = Q_0 + Q_1$$

on obtient

$$P_0 \cdot Q_1 + P_1 \cdot Q_0 = R \cdot S - P_0 \cdot Q_0 - P_1 \cdot Q_1$$

et il n'est plus nécessaire de calculer que 3 multiplications.

Écrire une multiplication de Karatsuba. Quel est sa complexité?

### 3 Induction vs itération: Fibonacci

On rappelle la suite de Fibonacci définie par

$$F(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ F(n-1) + F(n-2) & \text{sinon} \end{cases}$$

- Écrivez une fonction récursive `fib_rec` calculant le terme d'indice  $n$  de la suite de Fibonacci. Prouvez sa terminaison, sa correction, et calculez sa complexité.
- Écrivez une fonction itérative `fib_iter` calculant le terme d'indice  $n$  de la suite de Fibonacci. Prouvez sa terminaison, sa correction, et calculez sa complexité.