Travaux Dirigés d'algorithmique n°7

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Révisions

Nous abordons dans cette feuille de révisions les tableaux, les listes et le codage IEEE. Cette feuille contient entre autres des exercices donnés à de précédentes sessions d'examens.

▸ **Exercice 1.** *Écrire une fonction qui prend en argument un tableau de N entiers (N étant une constante) et qui modifie le tableau de façon à mettre les entiers pairs avant les entiers impairs. Par exemple, la fonction modifiera le tableau* `7 4 7 8 4 6 3 9 6` *en* `4 8 4 6 6 7 7 3 9` *ou bien encore en* `8 6 4 4 6 7 3 9 7` *: ce qui compte est que les nombres pairs se trouvent finalement avant les impairs. Calculez la complexité de votre fonction.*

▸ **Exercice 2.** ***(Juin 97)*** *Les listes chaînées d'entiers peuvent être représentées par la structure :*

```
typedef struct Cellule {
          int entier;
          struct Cellule * Suivant; } StrCellule;
typedef StrCellule * ListeEntier;
```

- *Écrire une fonction* `PremVersDern` *qui prend en argument une liste* `lc` *et qui place la première cellule en dernière position de la liste ;*
- *écrire une fonction* `EstlePlusPetit` *prenant en argument une liste d'entiers* `lc`*, et un entier* `n`*, et qui teste si* `n` *est plus petit (au sens large) que tous les éléments de la liste* `lc` *;*
- *en déduire une fonction de tri (récursive) prenant en argument une liste* `lc` *et qui effectue un tri sur la liste* `lc` *;*
- *évaluer la complexité de cette dernière fonction.*

▸ **Exercice 3.** *Donner un type de données pour représenter des matrices d'entiers à deux dimensions. Écrire deux fonctions, dont vous donnerez les complexités, qui retournent respectivement la somme et le produit de deux matrices passées en argument si possible, la matrice vide sinon.*

▸ **Exercice 4.** ***(Janvier 98)*** *Les listes chaînées d'entiers peuvent être représentées par la structure :*

```
typedef struct Cellule {
          int entier;
          struct Cellule * Suivant; } StrCellule;
typedef StrCellule * ListeEntier;
```

- *Écrire une fonction* `Construct` *prenant en argument un entier* `N` *et qui retourne la liste des entiers compris entre 2 et* `N` *(dans l'ordre croissant) ;*
- *écrire une fonction* `Elimine` *prenant en argument une liste d'entiers* `l` *et un entier* `n` *et qui supprime dans* `l` *tous les multiples de* `n` *;*
- *réécrire la fonction précédente* `ElimineBis` *sans utiliser ni multiplications, ni divisions (on supposera que la liste est triée).*

On veut maintenant implémenter sur des listes chaînées l'algorithme du crible d'Eratosthène. Cet algorithme permet de calculer les nombres premiers inférieurs à une borne fixée `N`*. On part de l'ensemble de tous les entiers inférieurs à* `N`*. On supprime tous les multiples de* `2` *(sauf* `2` *qui est premier). Le plus petit nombre* `>2` *qui reste est alors* `3` *qui est premier. On supprime alors tous ses multiples (sauf lui même). On recommence la même opération avec le plus petit nombre* `>3` *restant et ceci jusqu'à ce que la liste ne contienne que des nombres premiers.*

- *Déduire des questions précédentes une fonction* `Crible` *prenant en argument un entier* `N` *et qui retourne la liste des entiers premiers inférieurs à* `N`*.*

▸ **Exercice 5.** *Donnez en binaire le codage IEEE de* $(448)_{10}$ *et* $(0,625)_{10}$*. Quel nombre en base 10 est représenté par le codage IEEE* $(50D00000)_{16}$ *?*