Travaux Dirigés d'algorithmique n°9

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Révisions

Cette feuille de travaux dirigés rassemble des exercices donnés lors de précédents examens.

▸ **Exercice 1. (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 ;
```

- *Ecrire 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 2. (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 ;
```

- *Ecrire 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 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 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 3.** *On considère un tableau d'entiers $T = (t_0, t_1, t_2, \ldots, t_{n-1})$. On aimerait connaitre la plus longue suite croissante d'éléments consécutifs du tableau T. Par exemple, pour le tableau $T = (1, 2, 1, 3, 4, 5, 6, 2, 1, 4, 2, 2)$, la plus longue suite croissante commence à l'indice 2 et est de longueur 5.*

1. *Ecrire un algorithme qui calcule la longueur de la suite de nombres croissants commençant à l'indice i du tableau ;*
2. *en déduire un algorithme qui résoud le problème énoncé. On demande en sortie la longueur du plus long plateau, ainsi que l'indice de début de ce plateau.*
3. *Traduire cet algorithme en un programme C complet.*