■ T.D. 2 ■
La complexité

I - Divers

Exercice 1.

```
let g n =
  let m = ref 2 in
  for i = 1 to n do
    m := !m * !m
  done;
  !m
;;
```

1. Que calcule cette fonction ?

2. Déterminer sa complexité.

Exercice 2. On suppose que les entrées des programmes sont des tableaux de n entiers. Pour évaluer la complexité, on considère le nombre de comparaisons effectuées entre 2 éléments.

1. Écrire une fonction `maxi t` qui retourne le plus grand élément du tableau `t`. Quelle est sa complexité ? Peut-on faire mieux ?

2. Écrire une fonction `maxi_mini t` naïve qui retourne le couple contenant la valeur maximale et la valeur minimale de `t`. Quelle est sa complexité ?

3. On propose d'améliorer l'algorithme précédent en décomposant le traitement du tableau en trois phases :

(i). On compare par paire les élémnents de l'ensemble. On sépare, dans les cases d'indice impair, les plus grands éléments des plus petits.

(ii). On recherche le minimum parmi les plus petits éléments.

(iii). On recherche le maximum parmi les plus grands éléments.

a) Écrire une fonction `min_max` qui utilise l'algorithme précédent.

b) Déterminer la complexité, en nombre de comparaisons de cet algorithme.

c) Montrer que cet algorithme est optimal.

Exercice 3. Les entrées sont des tableaux de n éléments (non nécessairement tous distincts). Étant donné un élément e, on note $\sharp(e)$ le nombre d'occurences de e dans le tableau T. L'élément e est majoritaire si $\sharp(e) > n/2$.

1. Écrire une fonction `sharp t e` qui calcule $\sharp(e)$.

2. En déduire une fonction `existe_maj t` qui détermine si le tableau `t` contient un élément majoritaire.

3. Déterminer sa complexité en nombre de comparaisons d'éléments.

II - Tris

Exercice 4. Écrire une fonction `tri_borne t` qui, étant donné un tableau `t` d'entiers compris entre 0 et 9, retourne, en temps linéaire par rapport à la longueur de `t`, le tableau `t` trié.

Exercice 5. On suppose que les entrées sont des tableaux de n entiers compris entre 0 et k. On suppose que $k = O(n)$. Écrire une fonction `tri_lineaire` qui permet de trier ces tableaux en temps linéaire.