

# Structures de données

## TP : Tableaux et temps d'exécution

### Exercice 1 : Recherche du maximum

1. Écrivez un programme en C qui implémente un algorithme de recherche du maximum dans un ensemble de  $n$  entiers. Testez-le avec  $n = 5$ , puis avec  $n = 900\,000\,000$  (dans le cas où  $n$  est très grand, vous pouvez remplir le tableau en utilisant la fonction `rand()`).
2. Quel est le nombre de comparaisons effectuées dans votre algorithme en termes de  $n$  ?
3. Est-ce que votre algorithme est optimal ?

### Exercice 2 : Tri à bulles

Le tri à bulles s'effectue en  $n-1$  étapes (pour un tableau de  $n$  éléments). A la  $i$ ème étape on balaye le tableau en partant de la fin jusqu'à la case  $i$  et à chaque fois que l'élément courant est plus petit que son prédécesseur, on échange leur position. Cette méthode a pour effet de faire remonter au fur et à mesure les bulles les plus légères vers la surface.

- 1) Implémentez en langage C le tri à bulles.
- 2) Calculez et affichez le nombre de comparaison effectuées par ce dernier.
- 3) Comparez le résultat obtenu avec celui du cours.

### Exercice 3 : Recherche dichotomique

On considère un tableau  $A$  de  $n$  éléments, que l'on suppose trié en ordre croissant. On définit l'algorithme suivant (on supposera que *clé* est dans le tableau et on recherchera la première occurrence de cette valeur)

```
fonction Cherche_Dich_Ilt( $A$  : tableau;  $n$ ,  $clé$  : entier) : entier
 $d \leftarrow 1$ ;  $f \leftarrow n$ ;  $trouve \leftarrow \text{faux}$ 
répéter
     $i \leftarrow \lfloor (d + f) / 2 \rfloor$  // Partie entière inférieure de  $(d + f) / 2$ 
    si  $A[i] = clé$ 
        alors
             $trouve \leftarrow \text{vrai}$ 
        sinon
            si  $A[i] < clé$  alors  $d \leftarrow i + 1$  sinon  $f \leftarrow i - 1$  fin si
    fin si
jusqu'à  $trouve$ 
retourner( $i$ )
```

1. Implémentez et testez cet algorithme sur le tableau suivant avec  $clé = 30$ .

1	7	8	9	12	15	18	22	30	31
---	---	---	---	----	----	----	----	----	----

2. On suppose que le tableau  $A[1..n]$  contient  $n = 2^k$  éléments (où  $k$  est un entier positif). Combien d'itérations l'algorithme effectuera-t-il au maximum ? En déduire la complexité (en  $O$ ) de l'algorithme.