

## TD17 : Complexité en moyenne

### Exemple simple

On considère la fonction suivante :

```
#include <stdbool.h>

/** t est un tableau de n booléen
 * on cherche l'indice de la première case contenant true,
 * la fonction renvoie n si aucun case ne contient true */
int premier_un(bool *t, int n){
    for(int i=0; i<n; i=i+1){
        if(t[i]){
            return i;
        }
    }
    return n;
}
```

On considère comme opération élémentaire l'accès à une case d'un tableau.

**Question 1 :** Quelle est la complexité moyenne de cette fonction en fonction de la taille du tableau donné en entrée?

### Autour du tri rapide

**Question 2 :** Donner l'arbre d'appel de l'algorithme de tri rapide sur l'entrée suivante :

[7, 27, 25, 16, 0, 13, 16, 18, 19, 12, 23, 16, 1, 15, 10].

(on prendra le premier élément comme pivot à chaque étape)

On souhaite sélectionner l'élément de rang  $k$  dans un tableau de taille  $n$  ( $0 \leq k < n$ ).

**Question 3 :** Si on commence par trier le tableau, quelle est la complexité temporelle pour trouver l'élément de rang  $k$  dans le meilleur des cas? Dans le pire des cas?

**Question 4 :** On suppose que le tableau n'est pas trié. Proposer un algorithme basé sur le tri par sélection pour résoudre ce problème. Quelle est sa complexité temporelle en fonction des paramètres  $k$  et  $n$  dans le meilleur des cas? Dans le pire des cas? Est-ce mieux ou moins bien que l'algorithme précédent?

**Question 5 :** On suppose que le tableau n'est pas trié. Proposer un algorithme basé sur le tri rapide. Quelle est sa complexité temporelle en fonction des paramètres  $k$  et  $n$  dans le meilleur des cas? Dans le pire des cas? Est-ce mieux ou moins bien que les algorithmes précédents? [Si vous avez trouvé le bon algorithme, il s'agit de l'algorithme de sélection rapide, aussi connu sous le nom de *quickselect*.]

**Question 6 :** Montrer que la complexité en moyenne de l'algorithme de sélection rapide est en  $\Theta(n)$ .