

Chapitre 10

Temps d'exécution dans le pire des cas

Hypothèse sur les opérations informatiques simples

Chaque opération individuelle

- opération arithmétique (+, −, ×, /)
- comparaison
- affectation
- accéder à un élément dans un tableau
- appel ou retour d'une fonction
- ...

se fait en **temps constant** et indépendant de la taille des données. Chaque opération ne prend pas le même temps, mais cette durée est la constante : accéder à un élément d'un tableau ne dépend pas de la taille du tableau !

Exercice 27 (exercice)

1. Ecrire une fonction `premiere_valeur(T, x)` qui prend en argument un tableau de valeur `T` et qui renvoie -1 si `x` n'est pas dans `T` et l'indice de la première fois où `x` apparaît dans `T` sinon.
2. Si `x` est la première valeur de `T`, comment peut-on qualifier le temps d'exécution de votre fonction ?
3. Si `x` n'est pas dans `T`, comment peut-on qualifier le temps d'exécution de votre fonction ?

```
def premiere_valeur(T,x) :  
    for i in range(len(T)) :  
        if x == T[i] :  
            return i  
    return -1
```

Si `x` est la première valeur de `T`,

- appel de la fonction en temps constant
- affectation `i=0` en temps constant
- comparaison `i<len(T)` en temps constant
- comparaison `x==T[0]` en temps constant
- retour de fonction en temps constant

et qui fait que le temps d'exécution est globalement en temps constant.

Si x n'est pas dans T ,

- appel de la fonction en temps constant
- affectation $i = \dots$ en temps constant $\times n$
- comparaison $i < \text{len}(T)$ en temps constant $\times n$
- comparaison $x == T[0]$ en temps constant $\times n$
- retour de fonction en temps constant

et qui fait que le temps d'exécution est globalement en n .

Conclusion : on a ici un exemple d'un algorithme qui s'exécute dans le meilleur des cas en temps constant et dans le pire des cas en n .

Notation $\mathcal{O}()$

On utilise la notation $\mathcal{O}(f(n))$ pour qualifier le taux de croissance dans le **pire des cas**.

Donc ici, la fonction `premiere_valeur(T,x)` est une fonction en $\mathcal{O}(n)$ où n est la taille du tableau T .

taux de croissance	dans le pire des cas
temps constant	$\mathcal{O}(1)$
logarithmique $\lg n$	$\mathcal{O}(\lg n)$
linéaire	$\mathcal{O}(n)$
$n \lg n$	$\mathcal{O}(n \lg n)$
n^2	$\mathcal{O}(n^2)$
n^3	$\mathcal{O}(n^3)$
exponentiel	$\mathcal{O}(2^n)$