

# Exercices

## Exercice 1 :

On considère le tableau  $T = [2,3,5,7,9,4]$ .

1. Appliquez les algorithmes de tris par sélection et par insertion à ce tableau. Vous complèterez les deux tableaux ci-dessous pour détailler chaque itération.

Tri par sélection	T avant l'itération	T après l'itération	Nombre de comparaisons nécessaires	Nombre d'échanges nécessaires
Itération 1	[2,3,5,7,9,4]			
Itération 2				
...				

Tri par insertion	T avant l'itération	T après l'itération	Nombre de comparaisons nécessaires	Nombre de décalages nécessaires
Itération 1	[2,3,5,7,9,4]			
Itération 2				
...				

2. Comparez le nombre total de comparaisons nécessaires de ces deux algorithmes. Lequel semble le plus efficace ?

## Exercice 2 :

En supposant que le tri par sélection prend un temps directement proportionnel à  $n^2$  et qu'il prend 6,8 secondes pour trier 16 000 valeurs, calculez le temps qu'il faudrait pour trier un million de valeurs avec ce même tri par sélection.

## Exercice 3 :

Ecrire une fonction `est_trie(t)` qui renvoie `True` si le tableau `t` est trié par ordre croissant et `False` sinon.

## Exercice 4 :

Que renvoie chacune des instructions suivantes ?

1. `sorted([10, 2, 3, 21, 7])`
2. `sorted(['10', '2', '3', '21', '7'])`

## Exercice 5 :

1. Que vaut le tableau `t` après les instructions suivantes ? Justifiez.

```
t = [2, 1, 4, 7]
t.sort()
t[1] = 3
```

2. Que vaut le tableau `t` après les instructions suivantes ? Justifiez.

```
t = [2, 1, 4, 7]
sorted(t)
t[1] = 3
```

### **Exercice 6 :**

1. A quoi sert la fonction `f` suivante ?
2. Donnez un invariant de boucle pour cette fonction.

```
def f(t):
    """t est un tableau d'entiers"""
    s = 0
    for i in range(len(t)):
        s = s + t[i]
    return s/len(t)
```

### **Exercice 7 :**

1. Donnez un invariant de boucle pour la fonction suivante qui calcule  $x^n$ .
2. L'algorithme termine-t-il ? Justifiez.

```
def puissance(x, n):
    r = 1
    for i in range(n):
        r = r * x
    return r
```

### **Exercice 8 :**

Donnez un variant justifiant la terminaison de l'algorithme suivant qui renvoie le quotient et le reste de la division de `a` par `b` (avec `a` et `b` non nuls).

```
def division_euclidienne(a, b):
    q = 0
    r = a
    while r >= b:
        q = q + 1
        r = r - b
    return q, r
```

### **Exercice 9 :**

On recherche la valeur 9 par dichotomie dans le tableau `t` suivant.

```
t = [1, 2, 2, 5, 6, 6, 7, 9, 9, 10, 10, 13, 13, 15]
```

1. Complétez le tableau d'évolution des variables au cours de l'algorithme de recherche dichotomique.

	$g \leq d ?$	$m$	$T[m]$	Moitié à conserver (droite ou gauche ou fin ?)	$g$	$d$
<b>Avant</b> l'itération 1						
<b>Après</b> l'itération 1						
<b>Après</b> l'itération 2						
...						

2. Quelle est la valeur renvoyée par l'algorithme ?

### **Exercice 10 :**

On recherche la valeur 3 par dichotomie dans le tableau  $t$  suivant.

$$t = [1, 1, 2, 4, 5, 6, 6, 7, 9, 10, 10, 13]$$

1. Complétez le tableau d'évolution des variables au cours de l'algorithme de recherche dichotomique.

	$g \leq d ?$	$m$	$T[m]$	Moitié à conserver (droite ou gauche ou fin ?)	$g$	$d$
<b>Avant</b> l'itération 1						
<b>Après</b> l'itération 1						
<b>Après</b> l'itération 2						
...						

2. Quelle est la valeur renvoyée par l'algorithme ?

### **Exercice 11 :**

Combien de valeurs sont examinées lors d'un appel à `recherche_dichotomique([0, 1, 1, 2, 3, 5, 8, 13, 21], 7)` ?

### **Exercice 12 :**

Donnez un exemple d'exécution de la fonction `recherche_dichotomique` où le nombre de valeurs examinées est exactement 5.

### **Exercice 13 :**

Quand on joue au nombre mystère avec un nombre entier compris entre 1 et 100, combien faut-il d'essais dans le pire des cas si l'on joue de façon optimale ?

### **Exercice 14 :**

Quel est le nombre maximal de tours de boucle effectués par l'algorithme de recherche dichotomique dans un tableau trié de taille 60 000 000 ? de taille 7 500 000 000 ?

---

### **Ressources :**

- Documents ressources du DIU EIL, Université de Nantes.
- *Numérique et Sciences Informatiques*, T. BALABONSKI, S. CONCHON, J.-C. FILLIATRE, K. NGUYEN, Ellipses.