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 n² 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.

 Germain BECKER & Sébastien POINT, Lycée Mounier, ANGERS

```
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
```

Testons ce que vaut la variable r au cours des itérations pour puissance(3, 4) donc x = 3 et n = 4.

Avant l'itération d'indice **0** : r = 1 (=3^0) Après l'itération d'indice 0 : r = 1*3 =3	Avant l'itération 0, r = 3^**0**
Avant l'itération d'indice **1** : r = 3 (=3^1) Après l'itération d'indice 1 : r = 3 * 3 = 9	Avant l'itération 1, r = 3^**1**
Avant l'itération d'indice **2** : r = 9 (=3^2) Après l'itération d'indice 2 : r = 9 * 3 = 27	Avant l'itération 2, r = 3^**2**
Avant l'itération d'indice **3** : r = 27 (=3^3) Après l'itération d'indice 3 : r = 27 * 3 = 81	Avant l'itération 3, r = 3^**3**

On peut généraliser cela pour trouver notre invariant de boucle :

« Avant l'itération i, r = x^i »

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 <= d ?	m	T[m]	Moitié à conserver (droite ou gauche ou fin ?)	g	d
Avant l'itération 1					0	13
Après l'itération 1	True	6 car (0+13)//2 = 6	7	Droite (car 9 > 7)	7	13
Après l'itération 2	True	10 car (8+13)//2=10	10	Gauche (car 9 < 10)	7	9
Après l'itération 3	True	8 car (7+9)//2=9	9	Fin (car on a trouvé 9 !)		

2. Quelle est la valeur renvoyée par l'algorithme ? 8 = une position de 9 dans T

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 <= d ?	m	T[m]	Moitié à conserver (droite ou gauche ou fin ?)	g	d
Avant l'itération 1					0	11
Après l'itération 1	True	5	6	Gauche (car 3 < 6)	0	4
Après l'itération 2	True	2	2	Droite (car 3 > 2)	3	4
Après l'itération 3	True	3 Car (3+4)//2=3	4	Gauche (car 3 < 4)	3	3
Après l'itération 4	True	3 Car (3+3)//2=3	4	Gauche (car 3 < 4)	3	2
	False donc FIN					

2. Quelle est la valeur renvoyée par l'algorithme ? `None` car la valeur 3 n'a pas été trouvée dans le tableau.

Exercice 11 :

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

1ère itération : [0,1,1,2,$\boxed{3}$,5,8,13,21]
2ème itération : [~~0,1,1,2,3~~,5,$\boxed{8}$,13,21]
3ème itération : [~~0,1,1,2,3~~,$\boxed{5}$,~~8,13,21~~]

Les valeurs suivantes sont donc examinées : 3, 8, 5 cela en fait trois !

Exercice 12 :

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

On chercher donc une recherche dichotomique nous conduisant à 5 itérations. Or, si le tableau contient 15 valeurs ou moins, le nombre d'itérations sera au maximum égal à 4 (car $2^5 = 16$). On doit donc avoir un tableau d'au moins 16 éléments.

Si on choisit un tableau de 16 éléments et que l'on souhaite qu'il y ait 5 itérations, alors :

- Soit le tableau ne contient pas la valeur cherchée
- Soit la valeur cherchée est trouvé à la 5ème (et dernière) itération

Exemple 1 :
`recherche_dichotomique(1, [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])`

1ère itération : [0,0,0,0,0,0,0,$\boxed{0}$,0,0,0,0,0,0,0,0]
2ème itération : [~~0,0,0,0,0,0,0,0~~,0,0,0,$\boxed{0}$,0,0,0,0]
3ème itération : [~~0,0,0,0,0,0,0,0,0,0,0,0~~,0,$\boxed{0}$,0,0]
4ème itération : [~~0,0,0,0,0,0,0,0,0,0,0,0,0,0~~,$\boxed{0}$,0]
5ème itération : [~~0,0,0,0,0,0,0,0,0,0,0,0,0,0,0~~,$\boxed{0}$]

Exemple 2 :

`recherche_dichotomique(2, [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2])`

1ère itération : [1,1,1,1,1,1,1,$\boxed{1}$,1,1,1,1,1,1,1,2]
2ème itération : [~~1,1,1,1,1,1,1,1~~,1,1,1,$\boxed{1}$,1,1,1,2]
3ème itération : [~~1,1,1,1,1,1,1,1,1,1,1,1~~,1,$\boxed{1}$,1,2]
4ème itération : [~~1,1,1,1,1,1,1,1,1,1,1,1,1,1~~,$\boxed{1}$,2]
5ème itération : [~~1,1,1,1,1,1,1,1,1,1,1,1,1,1,1~~,$\boxed{2}$]

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 ?

7 essais !

En effet : $\frac{100}{2^6} \simeq 1{,}56 > 1$ et $\frac{100}{2^7} \simeq 0{,}78 < 1$.

1ère itération : on cherche dans un tableau de taille 100
2ème itération : on cherche dans un tableau de taille au plus 50 (100//2=50)
3ème itération : on cherche dans un tableau de taille au plus 25 (50//2=25)

4ème itération : on cherche dans un tableau de taille au plus 12 (25//2=12)
5ème itération : on cherche dans un tableau de taille au plus 6 (12//2=6)
6ème itération : on cherche dans un tableau de taille au plus 3 (6//2=3)
7ème itération : on cherche dans un tableau de taille au plus 1 (3//2=1)

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 ?

Pour un tableau de 60 000 000 valeurs : $\boxed{\text{26 tours de boucle max}}$!
En effet : $\frac{60\,000\,000}{2^{25}} > 1$ et $\frac{60\,000\,000}{2^{26}} \leq 1$ il faut donc diviser par deux 26 fois successivement pour passer de 60 000 000 à au plus une valeur.

Pour un tableau de 7 500 000 000 valeurs : $\boxed{\text{33 tous de boucle max}}$!
En effet : $\frac{7\,500\,000\,000}{2^{32}} > 1$ et $\frac{7\,500\,000\,000}{2^{33}} \leq 1$ il faut donc diviser par deux 33 fois successivement pour passer de 7 500 000 000 à au plus une valeur.

Ressources :

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