
Connaissance du cours

1. Donner la définition de :
 - recherche linéaire
 - recherche dichotomique
 - algorithme de tri
 - complexité. En particulier, Comment mesure t-on l'efficacité d'un algorithme?
 - Que signifie: complexité linéaire / logarithmique / quadratique ?
 2. Expliquer avec des mots le principe du tri par sélection.
 3. Expliquer avec des mots le principe du tri par insertion
 4. Donner l'algorithme de recherche linéaire, écrit soit en langage naturel, soit en langage Python
 5. Expliquer en quoi la recherche d'un mot dans un dictionnaire est une recherche dichotomique
-

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 :

1. Ecrire une fonction **est_trie(t)** qui renvoie **True** si le tableau t est trié par ordre croissant et **False** sinon.
2. On dispose d'une fonction appelée `tri_select`. Cette fonction va réaliser le tri d'une liste selon la méthode de tri par sélection. Comment pourrait-on utiliser cette fonction pour réaliser un test de POST condition?
3. Comment pourrait-on utiliser la fonction **est_trie** pour réaliser pour réaliser un test de PRE condition? Ce test devra faire sortir de la fonction de tri, sans rien faire (puisque la liste est déjà triée).

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 :

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.
2. Quelle est la valeur renvoyée par l'algorithme ? ($i_{\min} \Leftarrow g$, $i_{\max} \Leftarrow d$, $mid \Leftarrow m$)

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

Exercice 7 :

Combien de valeurs sont examinées par l'instruction `if t[m] == m` lors d'un appel à

`recherche_dichotomique([0,1,1,2,3,5,8,13,21], 7)` ?

Exercice 8 :

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

Exercice 9 :

1. A quoi sert la fonction `f` suivante ?
2. Quel est le nombre d'opérations (affectations, additions, divisions, ...) effectuées lors d'un appel à : `f([0, 10, 20, 90, 100])` ?

```
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 10 :

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 11 :

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 ?