

Spécialité NSI
Classe de première
Interrogation d'informatique n° 4
Sujet A
Mardi 1^{er} février 2022

Prénom :

Nom :

I Tri par insertion et par sélection

1. L'algorithme de tri par sélection possède une complexité :

- a) exponentielle b) linéaire c) logarithmique d) quadratique

2. On a trié par la méthode de tri par sélection un tableau L_1 de 5 000 éléments en 0,4 ms. On veut trier avec la même méthode et sur le même ordinateur un tableau L_2 de 30 000 éléments. Ce tri du tableau L_2 devrait prendre :

- a) 0,4 ms b) 0,7 ms c) 2,4 ms d) 14,4 ms

3. Si on compare l'algorithme de tri par insertion à l'algorithme de tri par sélection. On peut dire que le premier est :

- a) beaucoup plus efficace b) beaucoup plus efficace
c) d'une efficacité similaire d) trop différent pour être comparé

4. Écrire ci-dessous les *trois* étapes intermédiaires du tableau [21, 18, 19, 48, 28] lors du tri par insertion et du tri par sélection

Tri par sélection
[21, 18, 19, 48, 28]

[18, 19, 21, 28, 48]

Tri par insertion
[21, 18, 19, 48, 28]

[18, 19, 21, 28, 48]

5. La variable T1 contient un tableau non trié de nombres choisis aléatoirement. On exécute l'instruction suivante

`T2 = sorted(T1)`

Que contiennent alors les variables T1 et T2 ?

- a) La variable T1 contient un tableau trié et la variable T2 contient le tableau initial non trié.
b) Les variables T1 et T2 contiennent le tableau initial non trié.
c) Les variables T1 et T2 contiennent le tableau trié.
d) La variable T1 contient le tableau initial non trié et la variable T2 contient le tableau trié.

6. La variable T3 contient un tableau non trié de nombres choisis aléatoirement. On exécute l'instruction suivante

`T4 = T3.sort()`

Que se passe-t-il alors ?

- a) La variable T4 contient un tableau trié et la variable T2 contient None.
b) La variable T3 contient le tableau initial non trié et la variable T4 contient le tableau trié.
c) Un message d'erreur s'affiche.
d) La variable T3 contient le tableau trié et la variable T2 contient None.

II Recherche par dichotomie dans un tableau trié

1. L'algorithme de recherche par dichotomie possède une complexité :

- a) exponentielle b) linéaire c) logarithmique d) quadratique

2. On a écrit en python la fonction de recherche dichotomique en insérant des instructions d'affichage pour pouvoir suivre l'évolution des variables 'g' et 'd' :

```
def recherche_dicho(tab, val) :  
    g = 0  
    d = len(tab)-1  
    while g <= d :  
        print("g : ", g, "d : ", d)  
        m = (d + g)//2  
        if val < tab[m] :  
            d = m-1  
        elif val > tab[m] :  
            g = m+1  
        else :  
            return m  
    return None
```

Pour chacun des appels de fonctions suivants, indiquer ce qui va s'afficher dans la console :

a) `print(recherche_dicho([1, 7, 9, 11, 12, 14, 16, 20, 24], 16))`

|

b) `print(recherche_dicho([1, 7, 9, 11, 12, 14, 16, 20, 24], 5))`

|

3. On a passé 20 fois dans la boucle `while` pour faire une recherche dichotomique dans un tableau trié de taille N qui a renvoyé le résultat `None` Combien de fois au maximum passera-t-on dans la boucle `while` pour faire une recherche dichotomique dans un tableau trié de taille $10*N$?

- a) 20 fois b) 24 fois c) 200 fois d) 2400 fois