

Devoir surveillé n°4 – NSI
25/01/2022

Exercice 1 (2 points)

On considère l’algorithme en Python ci-dessous :

```
S=0
for i in range(n):
    x=i
    for j in range(1,n):
        x=x+j
    S=S+x
```

1. Quel est le nombre de sommes effectuées en fonction de n ? Expliquer.
2. Quelle est la complexité (linéaire, quadratique, etc.) de l’algorithme en fonction de n ?

Exercice 2 (8 points)

On considère ci-dessous une fonction écrite en langage Python.
Le paramètre `liste` est une liste de nombres flottants non vide (donc de longueur $n \geq 1$).

```
def fonction(liste):
    maxi = liste[0]
    n = len(liste)
    i = 1
    while i < n:
        if liste[i]>maxi:
            maxi = liste[i]
        i+=1
    return maxi
```

1. Que retourne `fonction([5,2.3,7,6])` ?
Détailler l’exécution de l’algorithme à l’aide du tableau suivant, à reproduire et compléter :

<code>i</code>	0	1	2	3	4
<code>i<n</code>	X				
<code>liste[i]</code>					
<code>liste[i]>maxi</code>	X				
<code>maxi</code>					

La ligne `i<n` est celle de la condition de la boucle `while`, il s’agit donc de remplir avec V (vrai) ou F (faux). Il en est de même pour la ligne `liste[i]>maxi`, condition du `if`.
Pour la première colonne ($i=0$), ces conditions ne sont pas testées, d’où les ‘X’.

2. Démontrer que la boucle de l’algorithme se termine.
3. Démontrer que la propriété « `maxi` est le maximum de `liste[0:i]` et $i \leq n$ » est un invariant de la boucle.
On rappelle que `liste[a:b]` est la sous-liste des éléments de `liste` d’indices a à $b-1$ si $b > a$, sinon c’est la liste vide.
4. Dédire des questions précédentes que la fonction retourne l’élément maximal de `liste`.

Devoir surveillé n°4 – NSI
27/01/2022

Exercice 1 (2 points)

1. On considère l’algorithme en Python ci-dessous :

```
s=0
for i in range(4):
    x=i
    for j in range(2,4):
        s=i+j
    s=s+i
```

Quel est le nombre de sommes effectuées ? Expliquer.

2. L’affirmation suivante est-elle vraie ?

« Un algorithme contenant une boucle itérative "Pour i allant de 1 à n" dans laquelle est appelée une fonction de coût linéaire (en fonction de n) a un coût au moins quadratique. »

Justifier la réponse. Une réponse sans justification ne sera pas comptée.

Exercice 2 (8 points)

On considère ci-dessous une fonction écrite en langage Python.
Le paramètre `liste` est une liste de nombres flottants.

```
def fonction(liste):
    res = 0
    n = len(liste)
    i = 0
    while i < n:
        res = res+liste[i]
        i+=1
    return res
```

1. Que retourne `fonction([5,2.3,7,6])` ?

Détailler l’exécution de l’algorithme à l’aide du tableau suivant, à reproduire et compléter :

i	0	1	2	3	4
i<n					
liste[i]					
res					

La ligne `i<n` est celle de la condition de la boucle `while`, il s’agit donc de remplir avec V (vrai) ou F (faux).

2. Démontrer que la boucle de l’algorithme se termine.

3. Démontrer que la propriété « `res = L[0]+...+L[i-1]` et `i ≤ n` » est un invariant de la boucle.

On considérera que la somme `L[0]+...+L[i-1]`, qui est vide si `i-1 < 0`, est nulle dans ce cas.

4. Dédurre des questions précédentes que la fonction retourne la somme des termes de la liste `liste`.