NSI – Terminale – Sujet « zéro » -

Exercice 2

Question 1 – Contenu de la pile Q

1. Combien de déplacements vers le bas comprend-il ?

Il comprend 2 déplacements

2. Justifier que tous les chemins allant de (0; 0) à (2; 3) ont une longueur égale à 6.

A chaque déplacement vers la droite et à chaque déplacement vers le bas on passe par une nouvelle case.

Pour déterminer la longueur d’un chemin, il faut donc additionner le nombre de cases correspondant au déplacement vers le bas, soit 2, au nombre de déplacement vers la droite, soit 3, et ajouter 1 pour la case de départ.

Donc tous les chemins allant de (0; 0) à (2; 3) ont une longueur égale à 1 + 2 + 3 = 6

Question 2 - Déterminer un chemin qui permet d’obtenir la somme maximale et la valeur de cette somme.

Chemins possibles	Somme maximale
$(0;0) \to (0;1) \to (0;2) \to (0;3) \to (1;3) \to (2;3)$	11
$(0;0) \to (0;1) \to (0;2) \to (1;2) \to (1;3) \to (2;3)$	10
$(0;0) \to (0;1) \to (0;2) \to (1;2) \to (2;2) \to (2;3)$	14
$(0;0) \to (0;1) \to (1;1) \to (1;2) \to (1;3) \to (2;3)$	9
$(0;0) \to (0;1) \to (1;1) \to (1;2) \to (2;2) \to (2;3)$	13
$(0;0) \to (0;1) \to (1;1) \to (2;1) \to (2;2) \to (2;3)$	12
$(0;0) \to (1;0) \to (1;1) \to (1;2) \to (1;3) \to (2;3)$	10
$(0;0) \to (1;0) \to (1;1) \to (1;2) \to (2;2) \to (2;3)$	14
$(0;0) \to (1;0) \to (1;1) \to (2;1) \to (2;2) \to (2;3)$	13
$(0;0) \to (1;0) \to (2;0) \to (2;1) \to (2;2) \to (2;3)$	16

Chemin qui permet d’obtenir la somme maximale : (0; 0); (1; 0); (2; 0); (2; 2); (2; 2); (2; 3)

La valeur de la somme de ce chemin est 16

Question 3

1. Tableau T’

4	5	6	**9**
6	**6**	8	10
9	10	**15**	16

2. Justifier que si j est différent de 0; alors : T’ [0] [j] = T [0] [j] + T’ [0] [j-1]

Nous savons que : $T'[0][j]=\sum_{i=0}^{j} T[0][i]$

Donc : $T'[0][j]=\sum_{i=0}^{j-1} T[0][i]+T[0][j]$

Puisque : $\sum_{i=0}^{j-1} T[0][i]=T'[0][j-1]$

Alors $T'[0][j]=T'[0][j-1]+T[0][j]$

Question 4 - Justifier que si i et j sont différents de 0; alors : T’ [i] [j] = T [i] [j] + max(T’ [i-1] [j]; T’ [i] [j-1]).

Si i et j sont non-nuls; il y a deux chemins amenant à la case (i; j).

Le premier provient de la case du dessus (i − 1; j); le second de la case de gauche (i; j − 1).

Or chacune de ces deux cases possèdent une valeur qui, selon la règle de construction du tableau T’, pour la case T’[i-1][j] correspond à la somme maximale pour tous les chemins possibles allant de (0; 0) à (i-1; j), et pour la case T’[i][j-1] correspond à la somme maximale pour tous les chemins possibles allant de (0; 0) à (i; j-1).

Et donc pour que l’élément T'[i][j] contiennent la valeur maximale pour tous les chemins allant de (0; 0) à (i; j), il faut ajouter à la valeur de T[i][j] la plus grande des valeurs entre les deux chemins menant à la case T'[i][j], c’est à dire soit la valeur de T'[i-1][j], soit celle de T'[i][j-1].

Question 5 - Fonction récursive ‘somme_max()’

1. Quel est le cas de base, à savoir le cas qui est traité directement sans faire appel à la fonction ‘somme_max()’ ? Que renvoie-t-on dans ce cas ?

Le cas de case qui est traité directement sans faire appel à la fonction somme_max(T,i, j) est le cas ou i et j valent ‘zéro’ (0).

Dans ce cas on renvoie T[0][0].

2. À l’aide de la question précédente, écrire en Python la fonction récursive somme_max .

```
def somme_max(T,i,j):
    if i == 0 and j == 0 :
        return T[0][0]
    elif i == 0:
        return T[0][j] + somme_max(T,0,j-1)
    elif j == 0:
        return T[i][0] + somme_max(T,i-1,0)
    else :
        return T[i][j] + max(somme_max(T,i-1,j), somme_max(T,i,j-1))
```

3. Quel appel de fonction doit-on faire pour résoudre le problème initial ?

```
max = somme_max(T,len(T)-1, len(T[0])-1)
```