ESIEE, unité d'algorithmique IN5i12 – mardi 20 janvier 2015 – rene.natowicz@eisee.fr

1) Spots et slots. La société *BrainStuffing*, société de diffusion de publicités télévisuelles, a acheté un intervalle de temps T (un *slot*) à la chaîne de télévision *TubeFeed*1 dans lequel elle diffusera des publicités (des *spots*) que ses clients lui ont confiées. Son carnet de commandes contient K spots. Le spot k, $0 \leq k < K$, est de durée $d(k)$. La diffusion du spot k apporte un gain $g(k)$ à la société *BrainStuffing*.

Justine, étudiante à l'Esiee, est en stage dans cette société. Lors d'une pause-café elle observe le travail de l'ingénieur commercial de *BrainStuffing* : il classe les spots par gains décroissants et les sélectionne dans cet ordre pour remplir le slot. À chaque étape il ajoute au slot le premier spot dont la durée est inférieure ou égale au temps encore disponible dans le slot. Le slot est de durée $T = 100$ secondes. Le carnet de commandes contient les dix spots suivants :

k	$d(k)$	$g(k)$	k	$d(k)$	$g(k)$
0	20	25	5	40	35
1	20	25	6	10	15
2	70	65	7	80	75
3	10	15	8	10	15
4	10	5	9	40	45

1) Quel sera le gain apporté par l'ingénieur commercial à la société *BrainStuffing* pour ce carnet de commandes ?

Justine a suivi l'excellent cours d'algorithmique de son école d'ingénieurs. Elle se dit qu'elle peut obtenir un gain supérieur. Ayant fini de faire les photocopies de la matinée, Justine se propose d'écrire une équation de récurrence qui lui permettra de calculer le gain maximum du carnet de commandes. Elle note $m(k, t)$ le gain maximum qu'elle peut obtenir pour un slot de durée t dans lequel la société *BrainStuffing* diffuserait des spots choisis dans le sous-ensemble des k premiers spots.

1) Base de la récurrence : $\forall t, 0 \leq t \leq T, m(0, t) =$ ...

2) Cas général : $\forall k, 1 \leq k \leq K, \forall t, 0 \leq t \leq T, m(k, t) =$ ....

3) Puis, Justine écrit le programme de calcul d'une matrice $M[0..K][0..T]$ de terme général $M[k][t] = m(k, t)$;

4) Elle donne la complexité de son programme ;

5) Elle écrit un programme qui affiche un sous-ensemble de spots dont la somme des gains est maximum. Voici L'exécution de son programme pour un slot de durée $T = 100$:

```
>>> main()
Spots=[(0,20,25),(1,20,25),(2,70,65),(3,10,15),(4,10,5),(5,40,35),(6,10,15),(7,80,75),(8,10,15),(9,40,45)]
T=100
Gain maximum = M[10][100] = 125
Un sous-ensemble de spot de gain maximum : (9,40,45)(6,10,15)(3,10,15)(1,20,25)(0,20,25)
>>>
```

2) Répartition optimale d'un stock. Une société commerciale dispose de K entrepôts et d'un stock S d'un produit à répartir entre les K entrepôts. Cette société connaît pour chaque entrepôt k, $0 \leq k < K$, et chaque quantité entière s de son stock, $0 \leq s \leq S$, le gain $g(k, s)$ qu'elle obtient en livrant à l'entrepôt k la quantité s de ce produit. Pour chaque entrepôt k le gain est nul si la quantité livrée est nulle $(g(k, 0) = 0)$ et la fonction de gain est non décroissante. Cette société veut connaître le gain total maximum qu'elle peut obtenir en répartissant au mieux son stock S sur ses K entrepôts. En notant $m(k, s)$ le gain maximum obtenu en répartissant un stock s sur les k premiers entrepôts, le gain total maximum est $m(K, S)$.

1) base de la récurrence : donner les valeurs $m(0, s)$ ($\forall s, 0 \leq s \leq S$) ;

2) cas général : donner les valeurs $m(k, s)$ ($\forall k, k \leq 1 \leq K$ et $\forall s, 0 \leq s \leq S$) ;

3) calculer une matrice $M[0..K][0..S]$ de terme général $M[k][s] = m(k, s)$;

4) donner la complexité de ce calcul ;

5) on note $l(k, s)$ la quantité de produit livrée au k-ième entrepôt (l'entrepôt de numéro $k - 1$) dans une répartition optimale d'un stock s sur les k premiers entrepôts. Modifier le programme afin de calculer à la volée une matrice $L[0..K][0..S]$ de terme général $L[k][s] = l(k, s)$;

6) écrire un programme d'affichage d'une répartition optimale du stock S sur les K entrepôts ;

7) choisir des fonctions de gain $g(s, k)$ et donner des exemples d'exécution.