ESIEE – Algorithmique 2 – IGI-2003 – TP1 – mercredi 5 et jeudi 6 mai 2015

René Natowicz – Ibrahim Alame – Arben Çela

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 du responsable marketing 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 le responsable marketing à la société *BrainStuffing* pour ce carnet de commandes ?

Ayant suivi l'excellent cours d'algorithmique de son école d'ingénieurs, Justine pense qu'elle peut obtenir un gain supérieur. Son travail de photocopie de la matinée terminé, Justine écrit 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.

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

3) cas général : $\forall k, 1 \leq k \leq K, \forall t, 0 \leq t \leq T, m(k,t) =$ ..... ;

4) Justine écrit le programme de calcul d'une matrice $M[0:K+1][0:T+1]$ de terme général $M[k][t] = m(k,t)$;

5) elle évalue la complexité de son programme ;

6) elle écrit un programme qui affiche un sous-ensemble de spots de gain total maximum ;

7) elle exécute son programme : il retourne la valeur d'une solution de gain total maximum et le sous-ensemble de spots correspondant.

Les rapports de TP sont le texte de la classe TP1.java (feuilles ci-jointes à compléter.) Ils sont à rendre au début de la prochaine séance de TP.

```
// Unité Algorithmique 2 (IGI-20034.) Séances des 5 et 6 mai 2015.
// Ces deux pages complétées constituent le rapport du TP.
// Noms du binôme :
// Groupe :
// Date :


// Question 1 (gain total obtenu par le responsable marketing) :

class TP1{ // spots et slots. Exemple d'exécution en fin de ce fichier.
    public static int[][] calculM(int[] D, int[] G, int T){
    /* D[0:K] de terme général D[k] = d(k), durée du spot k,
    G[O:K] de terme général G[k] = g(k), gain obtenu pour la diffusion du spot k,
    T est la durée du slot.

    Cette fonction retourne la matrice M[0:K+1][0:T+1] de terme général M[k][t] = m(k,t),
    gain total maximum que l'on peut obtenir en diffusant un sous-ensemble des k premiers
    spots dans un slot de durée t.

    Question 2 : Base de la récurrence : k=0.
         Pour tout t in [0:K+1], m(k,t) = ........................

    Question 3 : Cas général (récurrence) : k in [1:K+1].
         Pour tout t in [0:K+1], m(k,t) = ........................

    Question 4 : programme ci-dessous.
    */
    int K = D.length;
    int[][] M = new int[K+1][T+1];

        // base de la récurrence : m(0,t) = 0 pour tout t in [0:T+1]
        for (int t = 0 ; t <= T ; t++) ..................... ;

        // cas général : calcul par tailles k croissantes
        for (int k = 1 ; k <= K ; k++)
            // pour chaque taille k, calcul de toutes les valeurs m(k,t)
            for (int t = 0; t <= T ; t++) {
                M[k][t] = ................... ; // gain total max sans diffusion du k-e spot
                if (D[k-1] <= t){ // le k-e spot est diffusable (il n'est pas trop long.)
                    int mprime = .................; // gain total max si diffusion du spot
                    if (mprime > M[k][t]) // diffuser le k-e spot augmente le gain total
                        M[k][t] = mprime; // le gain total max est le gain avec diffusion du k-e spot.
                }
            }
        return M;
    }

    // Question 5 (complexité du programme) :


    // Question 6 (programme d'affichage d'une solution de gain total maximum)
    public static void afficherUneSolutionDeGainTotalMax(int[][] M, int[] D){
    /* L'ensemble de spots de gain total maximum diffusés dans le slot de durée T
    est décomposé en deux sous-ensembles :
    1) le sous-ensemble P(k,t) des spots du k-prefixe, diffusés dans le slot [0:t+1].
    2) le sous-ensemble S(k) des spots du k-suffixe.

    Propriété invariante I(k,t) : P(k,t) reste à afficher et S(k) a été affiché.
    Initialisation : k = ......, t = .........
    Arrêt : k = .........
    Progression : I(k,t) et k != K et M[k][t] = M[k-1][t] ==> I(......, ......)
    I(k,t) et k != K et M[k][t] > M[k-1][t] et ................... ==> I(......, .........)
    */









    }
```

```
75
76      public static void main(String[] args){
77          int[]   D = {20, 20, 70, 10, 10, 40, 10, 80, 10}, // durée de chaque spot
78                  G = {25, 25, 65, 15,  5, 35, 15, 75, 15} ; // gain si diffusion du spot
79
80          int K = D.length, // nombre de spots
81              T = 100; // durée du slot
82          System.out.print("Numéros des spots : \t"); afficherVecteur(range(K));
83          System.out.print("Durée de chaque spot  : "); afficherVecteur(D);
84          System.out.print("Gain si diffusion du spot : "); afficherVecteur(G);
85
86          int[][] M = calculM(D, G, T);
87          // System.out.println("Matrice des gains totaux maximum : "); afficherMatrice(M);
88
89          System.out.println("Gain total maximum = " + M[K][T]) ;
90          System.out.print("Une solution de gain total maximum : ");
91          afficherUneSolutionDeGainTotalMax(M, D);
92      }
93
94      public static void afficherVecteur(int[] V){
95          System.out.print("\t");
96          for (int i=0; i<V.length; i++) System.out.print(V[i]+"\t");
97          System.out.println();
98      }
99
100     public static int[] range(int K){ // retourne un tableau d'entiers R = [0:K]
101         int[] R = new int[K];
102         for (int i=0; i<K; i++) R[i] = i ;
103         return R;
104     }
105
106     public static void afficherMatrice(int[][] A){
107         int m = A.length, n = A[0].length;
108         for (int i = m-1; i > -1; i--){
109             for (int j = 0; j < n ; j++) System.out.print(A[i][j] + " ");
110             System.out.println();
111         }
112     }
113 }
114
115 // Question 7 (une solution optimale et sa valeur de gain total.)
116 /* Compilation, exécution : copier/coller le résultat de l'exécution de votre programme.
117 javac TP1corrige.java
118 $ java TP1corrige
119 Numéros des spots :         0   1   2   3   4   5   6   7   8
120 Durée de chaque spot  :     20  20  70  10  10  40  10  80  10
121 Gain si diffusion du spot : 25  25  65  15  5   35  15  75  15
122 .
123 .
124 .
125 .
126 .
127 .
128 $
129 */
130
```