

```

1 import java.util.Arrays;
2 public class Exercice6{
3     static final int plusInfini = Integer.MAX_VALUE;
4     /* Feuille 2, exercice 6 : trajet de coût minimum, il existe un trajet direct entre tout
5     couple de villes (i, j),  $i < j$ .
6     Equation de récurrence des valeurs  $m(j)$ , coût min. d'un trajet de 0 à j inclus.
7     base  $m(0) = 0$ , hérédité  $m(j) = \min \{m(i) + D_{ij}\}$  sur tous les i tels que  $0 \leq i < j$ 
8     */
9     static int[][] calculerMA(int[][] D){ int n = D.length;
10    /* calcule et retourne MA = {M, A}. où A = arg M */
11    int[] M = new int[n], A = new int[n];
12    M[0] = 0; // base (la valeur A[0] est quelconque)
13    // cas général  $m(j) = \max \{m(i) + D_{ij}\}$  sur tous les i tels que  $0 \leq i < j$ 
14    for (int j = 1; j < n; j++){ // pour j dans [1:n]
15        // calcul de  $m(j) = \min \{m(i) + D_{ij}\}$  sur tous les i tels que  $0 \leq i < j$ 
16        M[j] = plusInfini;
17        for (int i = 0; i < j; i++){ int mij = M[i]+D[i][j];
18            /* mij est le coût du chemin 0 -----m(i)-----> i --Dij--> j
19            Si mij est inférieur aux coûts minimum calculés jusqu'ici :
20            mettre à jour M[j] et A[j] */
21            if (mij < M[j]){
22                M[j] = mij;
23                A[j] = i; // A[j] est la dernière ville avant j sur le trajet de
24                // coût minimum de la ville 0 à la ville j incluses.
25            }
26        }
27    }
28    return new int[][] {M, A};
29 } // Forme du terme dominant du temps de calcul :  $\alpha \times n^2$  ( $n^2$  est "n au carré")
30 // car corps de boucle en temps majoré par une constante, exécuté
31 //  $1 + 2 + \dots + n-1$  fois =  $n \times (n-1) / 2$  fois
32
33 static void afficherTrajetMinimum(int[] M, int[] A, int[][] D, int j){
34     // Affiche un trajet de coût minimum de 0 à j inclus.
35     if (j==0){ // le trajet ne contient que la ville 0. L'afficher.
36         System.out.print(j);
37         return;
38     }
39     int aj = A[j]; // aj est la dernière ville avant j sur le trajet min de 0 à j.
40     // Le trajet min de 0 à j est : trajet min de 0 à aj puis direct aj -> j
41     // 1) afficher le trajet min. de 0 à aj
42     afficherTrajetMinimum(M, A, D, aj); // 0 -----m(aj)----->aj affiché
43     // 2) afficher "--D[aj][j]--> j" /
44     System.out.printf("--(%d)-->%d", D[aj][j], j);
45     // 0 -----m(aj)----->aj->D[aj][j]->j affiché (trajet coût min de 0 à j)
46 } // Pire cas : toutes les villes sont sur le trajet de coût minimum.
47 // Dans ce pire cas il y a n appels, chacun en temps majoré par une constante.
48 // Donc forme du temps de calcul dans le pire cas :  $\alpha \times n + \beta$ 
49
50 public static void main(String[] args){
51     { System.out.println("Exercice 6 : trajet de coût minimum");
52       final int inf = plusInfini;
53       int[][] D = { // les valeurs dij,  $i \geq j$ , sont quelconques. Ici : infini.
54           {inf, 40, 40, 90, 150}, // coûts des trajets directs de 0 à 1,2,3,4
55           {inf, inf, 50, 50, 100}, // de 1 à 2,3,4
56           {inf, inf, inf, 40, 100}, // de 2 à 3,4
57           {inf, inf, inf, inf, 50}, // de 3 à 4
58           {inf, inf, inf, inf, inf} // de 4 à ... aucun autre sommet
59       };
60       System.out.println("tableau D des coûts directs :");
61       afficherD(D);
62       int n = D.length;
63       System.out.println("Nombre de villes : " + n);
64       int[][] MA = calculerMA(D);
65       int[] M = MA[0], A = MA[1];
66       System.out.printf("M = %s\n", Arrays.toString(M));
67       System.out.printf("trajet de coût minmum de la ville 0 à la ville %d :\n",n-1);
68       afficherTrajetMinimum(M,A,D,n-1); System.out.println();
69       System.out.printf("Coût de ce trajet : %d\n", M[n-1]);
70       System.out.println();
71     }
72 } // end main()

```

```

73
74     static void afficherD(int[][] D){int m = D.length, n = D[0].length;
75         for (int i = m-1; i > -1; i--){ System.out.print(i + " : ");
76             for (int j = 0; j < n; j++)
77                 if (D[i][j] == plusInfini)
78                     System.out.print("inf"+" ");
79                 else
80                     System.out.print(D[i][j] + " ");
81             System.out.println();
82         }
83     }
84 } // end class
85
86 /* Compilation et exécution dans un terminal Unix
87
88 % javac Exercice6.java
89 % java Exercice6
90 Exercice 6 : trajet de coût minimum
91 tableau D des coûts directs :
92 4 : inf inf inf inf inf
93 3 : inf inf inf inf 50
94 2 : inf inf inf 40 100
95 1 : inf inf 50 50 100
96 0 : inf 40 40 90 150
97 Nombre de villes : 5
98 M = [0, 40, 40, 80, 130]
99 trajet de coût minmum de la ville 0 à la ville 4 :
100 0--(40)-->2--(40)-->3--(50)-->4
101 Coût de ce trajet : 130
102 %
103 */
104
105

```