141-INF4100-10
Devoir 4

Louise Laforest

Date de remise : Vendredi le 18 avril 2014 avant minuit

Le devoir peut être fait en équipe d'au plus deux personnes

DIRECTIVES

1. Vous devez remettre vos fichiers dans une archive `devoir4.zip` via le système **oto** dans la boîte **INF4100** de l'enseignant ℓ**aforest_**ℓ (toutes des lettres minuscules) à l'adresse suivante `http://oto.uqam.ca/application-web/connexion`
2. **Politique concernant les retards** : pénalité de $\frac{m}{144}$ points sur 100 où m est le nombre de minutes de retard (donne 10 points pour 24 heures).

Nous avons vu un **algorithme glouton** permettant de résoudre le problème de la **petite monnaie**. Cependant, la méthode ne donne pas toujours la solution optimale. Par exemple, si nous disposons de pièces de 1, 5, 11 et 25 unités et que nous devons faire la monnaie pour un montant de 15 unités, la solution gloutonne donnera comme solution une pièce de 11 unités et quatre pièces de 1 unité, soit cinq pièces. Cette solution n'est pas optimale puisque trois pièces de 5 unités permettent de rendre la monnaie. Ce problème peut aussi se résoudre par programmation dynamique. Soit n, le nombre de pièces distinctes et soit P, un tableau, dont les indices vont de 1 à n contenant la valeurs des n pièces. On suppose ici que nous disposons d'un nombre illimité de chaque type de pièce. Soit L le montant dont on veut obtenir la monnaie. Dans ce problème, les valeurs des pièces sont strictement positives. Le montant à remettre est un nombre entier non négatif. On assume que le tableau P est en ordre croissant.

Nous allons utiliser une matrice C de n lignes numérotées de 1 à n et $L+1$ colonnes numérotées de 0 à L où $C[i,j]$ contient le nombre minimal de pièces pour rendre la monnaie d'un montant j si l'on se limite seulement qu'aux pièces de valeurs P_1, P_2, ..., P_i. Si la monnaie ne peut être rendue, $C[i,j]$ sera égal à l'infini.

(a) Peu importe P, que vaut $C[1,j]$ pour $0 \leq j \leq L$?

(b) Peu importe P, que vaut $C[i,0]$ pour $1 \leq i \leq n$?

(c) Peu importe P, que vaut $C[i,1]$ pour $1 \leq i \leq n$?

(d) Supposons que $P = [1,5,11,25]$ et que $L = 20$, donnez le contenu du tableau C.

(e) Donnez une équation de **récurrence** pour $C[i,j]$. N'oubliez pas les conditions initiales (cas de base). *Aide* : Complétez ce qui suit (il existe d'autres réponses possibles).

$$C[i,j] = \begin{cases} \ldots & \text{si } j = 0, \\ \ldots & \text{si } i = 1 \text{ et } \ldots, \\ \ldots & \text{si } i = 1 \text{ et } \ldots, \\ \ldots & \text{si } j < P_i, \\ \ldots & \text{sinon.} \end{cases}$$

(f) Décrivez en pseudo-code un algorithme de **programmation dynamique** pour calculer tous les éléments de la dernière ligne du tableau C, c'est-à-dire les $C[n,j]$ pour $0 \leq j \leq L$. Votre algorithme ne doit utiliser **qu'un seul tableau à une dimension** de longueur $L+1$ et non pas un tableau à deux dimensions comme le tableau C.

(g) Donnez la **complexité** de votre algorithme en fonction de n et de L.

(h) Décrivez en pseudo-code un algorithme **glouton** capable de faire la monnaie en un nombre minimal de pièces une fois les $C[n,j]$ calculés, pour un montant quelconque $M \leq L$. Votre algorithme recevra donc en entrée la dernière ligne de la matrice C, le tableau P ainsi que le montant M.

(i) Donnez la **complexité** de l'algorithme trouvé en (h) en fonction de n.

Pour tester votre programme, celui-ci devra lire les données dans un fichier texte, appelé `monnaie.txt`, dont le format sera le suivant :

- première ligne : le nombre de pièces,
- deuxième ligne : le tableau P,
- troisième ligne : un nombre entier supérieur à 0,
- quatrième ligne : une suite de nombres entiers non négatifs dont la quantité est spécifiée par la ligne précédente correspondant à des montants pour lesquels on désire rendre la monnaie.

Tous les nombres seront séparés par un espace. **Veuillez respecter ce format.**
Votre programme devra lire ces données, faire les calculs nécessaires et écrire dans un fichier texte nommé `resultats.txt` le contenu du tableau P, le contenu de la matrice C calculée pour un montant $L = 20$ puis, pour chaque montant, la monnaie à rendre. Voici un exemple.

Fichier `monnaie.txt`

```
4
1 5 10 25
7
234 23 41 2 2605 45 117
```

Fichier `resultats.txt`

```
1 5 10 25
0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
0   1   2   3   4   1   2   3   4   5   2   3   4   5   6   3   4   5   6   7   4
0   1   2   3   4   1   2   3   4   5   1   2   3   4   5   2   3   4   5   6   2
0   1   2   3   4   1   2   3   4   5   1   2   3   4   5   2   3   4   5   6   2

Montant : nombre de pièces en ordre décroissant de valeur
234 : 9 0 1 4
23 : 0 2 0 3
41 : 1 1 1 1
2 : 0 0 0 2
2605 : 104 0 1 0
45 : 1 2 0 0
117 : 4 1 1 2
```

Vous devez me remettre un programme (listage + exécution) qui montre le bon fonctionnement des algorithmes trouvés en (f) et (h). Votre programme devra être écrit en Java, C++ ou Python. Vous devez indiquer avec quel compilateur votre programme a été compilé et sur quelle plateforme. Vous devez me remettre le source et, si possible, l'exécutable.

C'est à vous de me convaincre que votre code fonctionne et non à moi de vous convaincre que votre code ne fonctionne pas, le cas échéant.