Université de Rennes

Licence Informatique ISTIC
Méthodes Algorithmiques (MALG) *2023-2024*

Programmation dynamique

Exercice 1 (Gravir les marches)
On considère un escalier que l'on peut gravir par des enjambées d'amplitude 1 ou 2 marches. Notons par $N[k, n]$ le nombre de façons différentes de gravir n marches en k enjambées.

1. Trouvez les formules de récurrence qui permettent de calculer les valeurs de $N[k, n]$. *Suggestion: Considérez d'abord les cas extrêmes, tels que $k > n$ et $k = n$. On supposera naturellement qu'on ne redescend pas de marche.*

   Pour le cas général, remarquez que pour atteindre la marche n en k enjambées, il faut soit (a) avoir atteint la marche $n - 1$ en $k - 1$ enjambées puis effectuer une dernière enjambée de 1 marche, soit (b) avoir atteint la marche $n - 2$ en $k - 1$ enjambées puis effectuer une dernière enjambée de 2 marches.

2. Appliquez vos formules à un escalier de 8 marches.

3. Écrire l'algorithme qui permet de calculer $N[k, n]$.

4. Déduisez la complexité en temps et en mémoire pour calculer la valeur $N[k, n]$.

5. Donnez la formule définissant le nombre T_n de façons différentes de gravir n marches.

6. Exprimez T_n en fonction de T_{n-1} et T_{n-2}. À quelle suite célèbre correspond cette relation ?

Exercice 2 (Calcul du coefficient de convivialité)
Les n employés d'une société sont identifiés par les entiers $1, 2, \ldots, n$. L'organigramme des responsabilités dans la société est une arborescence dont la racine est le directeur de la société; chaque employé autre que le directeur est donc placé sous la responsabilité immédiate d'une seule personne : son supérieur hiérarchique.
Le service des relations humaines de la société a attribué un coéfficient entier $C(i) > 0$ ($i = 1, \ldots, n$), dit *de convivialité*, à chaque employé pour caractériser son comportement social. Le coéfficient est d'autant plus grand que l'employé est socialement apprécié.
La numérotation (l'identification) des employés est effectuée par niveaux de l'arborescence, on assure ainsi que tout employé possède un numéro plus grand que celui de son supérieur hiérarchique, le directeur ayant le numéro 1. Pour chaque employé i, $\mathrm{Emp}(i) \subseteq \{1, \ldots, n\}$ est l'ensemble des employés dont i est le supérieur hiérarchique direct.
Il est décidé d'organiser une fête à laquelle seront invités certains employés ; l'invitation doit respecter les deux règles suivantes:

(r1) un employé et son supérieur hiérarchique direct ne peuvent être invités simultanément;

(r2) le sous-ensemble des employés invités doit offrir une convivialité maximale.

Pour la Figure 1, le sous-ensemble $\{2, 3, 7, 8\}$ satisfait la règle **(r1)** mais sa convivialité de valeur 10, n'est pas maximale.
On veut calculer le tableau d'entiers $N[1, \ldots, n]$ tel que $N[i]$ est la convivialité maximale obtenue en n'invitant que des employés figurant dans la sous-hiérarchie dont i est la racine et en respectant la règles **(r1)**.

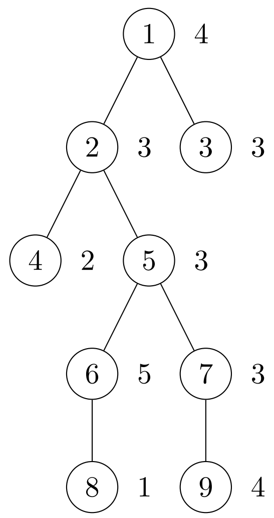

Figure 1: Un exemple de hiérarchie avec $n = 9$ employés. La convivialité de chaque employé est indiquée à droite du noeud correspondant.

1. Donnez la formule de récurrence qui permet de calculer $N[i]$ à partir de $C(i), deN[j]$ pour $j \in \text{Emp}[i]$, et de $N[k]$ pour $k \in \text{Emp}[j]$ et $j \in \text{Emp}[i]$ (i.e., les valeurs obtenues pour les personnes figurant dans la sous-hiérarchie de racine i).
2. Appliquez cette formule pour remplir le tableau $N[1 \ldots 9]$ pour l'exemple de la Figure 1, déterminez en même temps les ensembles $E[i]$ permettant d'obtenir ces valeurs optimales $N[i]$.
3. Écrivez un algorithme itératif qui effectue le remplissage de $N[1 \ldots n]$ et $E[1 \ldots n]$.
4. Donnez sa complexité temporelle.

Exercice 3 (Partition équilibrée)
On se donne un ensemble $E = \{e_1, \ldots, e_n\}$ de n entiers entre 0 et K.
On convient que pour tout $A \subseteq E$, $sum(A)$ désigne la somme des éléments de A.
On souhaite calculer une *partition équilibrée*, c'est à dire une partition de E en deux sous-ensembles E_1 et E_2 qui minimisent $|sum(E_1) - sum(E_2)|$.

3.1 *Soit $S = sum(E)/2$. On dit qu'un sous-ensemble $A \subseteq E$* minimise sa distance à S *si la valeur $S - sum(A)$ est minimale parmi tous les autres sous-ensembles possibles, c'est à dire si*

$$A = \operatorname*{argmin}_{M \subseteq E} \{S - sum(M)\}$$

Montrer que si on dispose d'un sous-ensemble A qui minimise sa distance à S, alors la paire A et $A' = E \setminus A$ est une solution à notre problème de partitionnement équilibré.

3.2 *Proposer un algorithme de programmation dynamique qui calcule la valeur $sum(A)$ d'un ensemble A qui minimise sa distance à S.*

3.3 *Donner un invariant de boucle qui servirait à montrer la correction de l'algorithme. Quelle est la complexité de votre algorithme ?*

3.4 *Et que peut-on dire d'un glouton qui trie par ordre décroissant et équilibre à la volée les deux ensembles ?*

Exercice 4 (Nombre de séquences triées)
Étant donnés deux nombre entiers $n \geq 1$ et $k \geq 1$, on désire calculer le cardinal $s_{n,k}$ de l'ensemble $S_{n,k}$ de séquences telles que de telles séquences sont :

(i) composées de n nombres, chacun compris entre 1 et k;

(ii) triées par valeurs croissantes.

1. A priori, de telles séquence peuvent comporter plusieurs fois le même entier. Ainsi,

$$S_{2,3} = \{< 1,1 >, < 1,2 >, < 1,3 >, < 2,2 >, < 2,3 >, < 3,3 >\}$$

   et on a $s_{2,3} = |S_{2,3}| = 6$.

   (a) Écrivez, en les justifiant, les formules de récurrence qui caractérisent $S_{n,k}$.
   (b) À l'aide de ces formules, déterminez les équations qui caractérisent les valeurs $s_{n,k}$.
   (c) Calculez les valeurs de $s_{n,k}$ pour $1 \leq k, n \leq 5$.
   (d) Proposez un algorithme qui étant donnés n et k retourne $s_{n,k}$. Quelle est sa complexité ?

2. On souhaite se limiter aux séquences triées :

   (iii) composées d'éléments distincts.

   On notera $R_{n,k} \subseteq S_{n,k}$ l'ensemble de séquences obéissant aux contraintes (i)-(iii), et on notera $r_{n,k} = |R_{n,k}|$ son cardinal.

   Ainsi, $R_{2,3} = \{< 1,2 >, < 1,3 >, < 2,3 >\}$ et $r_{2,3} = 3$.

   (a) Déterminez, en les justifiant, les formules de récurrence qui caractérisent $r_{n,k}$.
   (b) Déterminez les formules qui caractérisent les valeurs de $r_{n,k}$.

(c) Calculer $r_{n,k}$ pour $1 \leq n \leq k \leq 5$.

(d) Proposez un algorithme qui étant donnés n et k retourne $r_{n,k}$. Quelle est sa complexité ?

Exercice 5 (Empilement de boites)
On dispose de n de boites, et on convient que la $i$ème boite a une hauteur $h(i)$, une largeur $l(i)$ et une profondeur $p(i)$ (des nombres entiers). On veut empiler ces boites pour former une tour de hauteur maximale, sachant que l'on pourra empiler une boite i sur une boite j que si $l(j) > l(i)$ et $p(j) > p(i)$. Proposez une solution basée sur la programmation dynamique.
Invariants de boucle pour montrer la correction:

- Ligne 3 (boucle sur i) : pour tout $1 \leq i' < i$, $HTab[i'] = H(i')$ et $maxH = \max\{H(i'), 1 \leq i' < i\}$.
- Ligne 4 (boucle sur j): $HTab[i] = \max\{H(j'), 1 \leq j' < j$ et $l(j) > l(i)$ et $p(j) > p(i)\}$.

L'algorithme a donc un temps d'éxécution en $O(n^2)$ avec un espace mémoire de taille n.