

Examen d'Algorithmique et Programmation

Math 312

18 décembre 2008

Durée de l'examen : 3 heures

Tous les documents statiques sont autorisés.

Aucun outil dynamique (calculatrice, pda, téléphone...) n'est admis.

Le sujet comporte 3 pages. Il est composé de 5 exercices indépendants, qui peuvent être traités dans un ordre quelconque. L'exercice 1 doit être rédigé en C, les autres exercices peuvent être décrits dans un langage algorithmique de haut niveau. Le barème est donné à titre indicatif. La qualité de la présentation sera prise en compte dans l'évaluation de la copie.

1 Programmation en C [3 pts]

Écrire un programme en C qui demande à l'utilisateur un entier n et affiche les n premières lignes du triangle de Pascal.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

On rappelle que le triangle de Pascal est une façon de présenter et de calculer les coefficients binomiaux $\binom{i}{j}$ (parfois noté C_i^j) qui satisfont la relation $\binom{i}{j} = \binom{i-1}{j-1} + \binom{i-1}{j}$.

2 Tris [1,5 pts]

Étant donné un tableau de n nombres, on veut obtenir les i plus grands d'entre eux triés. Pour chacune des trois méthodes suivantes, donner la complexité en fonction de n et de i . Quelle est la meilleure méthode ?

- (a) Trier les n nombres et renvoyer les i plus grands.
- (b) Trouver le i -ème plus grand élément (à l'aide d'un algorithme linéaire), partitionner et trier les i plus grands.
- (c) Construire un tas avec les n nombres et extraire la racine i fois.

3 Algorithmes gloutons [4,5 pts]

La bibliothèque prévoit de refaire ses étagères. Elle comprend une collection de n livres b_1, b_2, \dots, b_n . Chaque livre b_i a une largeur w_i et une hauteur h_i . Les livres doivent être rangés dans l'ordre fixé (par valeur de i croissante) sur des étagères de largeur L .

1. On ne se soucie d'abord pas de la hauteur des livres. On cherche un algorithme glouton optimal pour ranger les livres sur les étagères en minimisant le nombre d'étagères (il existe). Proposez le meilleur algorithme que vous pouvez trouver.
2. Si c'est le cas, montrez que votre algorithme trouve la solution optimale. Sinon, retournez à la question précédente ! (ou passez à la suivante).
3. On peut maintenant régler la hauteur des étagères. Nous allons donc prendre en compte la hauteur des étagères. Nous cherchons maintenant à minimiser l'encombrement, défini comme la somme des hauteurs du plus grand livre de chaque étagère utilisée. Donnez un exemple où l'algorithme glouton précédent ne trouve pas la solution optimale.

4 Diviser pour régner [4 pts]

Soient $A[1..n]$ et $B[1..n]$ deux tableaux triés par ordre croissant. On cherche à trouver l'élément médian de ces deux tableaux (élément qui a autant d'éléments supérieurs stricts que d'éléments inférieurs ou égaux¹).

Proposez un algorithme diviser pour régner qui trouve le médian en $\mathcal{O}(\log n)$. Justifiez la complexité.

¹En fait, ceci est un raccourci. Le médian est un élément m tel qu'il existe une partition de l'ensemble des éléments en deux ensembles E_p et E_g telle que $|E_p| = |E_g|$, $\forall x \in E_p, x \leq m$ et $\forall y \in E_g, y \geq m$

5 Programmation dynamique [7 pts]

Cette année, la mère Noël s'est exceptionnellement proposée pour aider le père Noël. Ils vont donc devoir à eux deux passer par toutes les cheminées pour déposer leurs cadeaux. Ils aimeraient pouvoir réduire la longueur de leurs trajets. Le problème est le suivant : étant données un point de départ x_0 et n cheminées x_1, \dots, x_n , trouver deux chemins p_0, p_1, \dots, p_{n_p} et m_0, m_1, \dots, m_{n_m} avec $p_0 = m_0 = x_0$ et $\{x_1, \dots, x_n\} = \{p_1, \dots, p_{n_p}\} \cup \{m_1, \dots, m_{n_m}\}$ qui minimisent

$$\sum_{i=1}^{n_p} d(p_{i-1}, p_i) + \sum_{j=1}^{n_m} d(m_{j-1}, m_j)$$

où $d(x_i, x_j)$ désigne la distance entre x_i et x_j .

En fait, pour profiter du décalage horaire, le père Noël a l'habitude de voyager plutôt d'est en ouest, en partant de la ligne de changement de date. Pour simplifier le problème, nous allons donc supposer que ni le père Noël, ni la mère Noël ne pourront revenir vers l'est. De plus, nous supposons notre connaissance des positions des cheminées si précise que 2 cheminées n'ont jamais la même longitude².

1. Expliquer brièvement pourquoi il est intéressant de considérer que les x_i sont triés par longitude décroissante.
2. Supposons que le père Noël a visité en dernier la cheminée x_i et la mère Noël la cheminée x_j . La prochaine cheminée à visiter est la cheminée x_k avec $k = \max(i, j) + 1$. Nous nous trouvons dans un sous problème du problème principal. On désigne par $\mathcal{L}(i, j, [k..n])$ la longueur de la solution optimale de ce sous-problème. En particulier, $\mathcal{L}(0, 0, [1..n])$ est la longueur de la solution optimale du problème initial. Exprimez $\mathcal{L}(i, j, [k..n])$ en fonction des $\mathcal{L}(*, *, [k + 1..n])$ (remplacer les $*$ par des valeurs pertinentes).
3. Décrivez un algorithme de programmation dynamique qui permet de trouver la longueur optimale³ des chemins pour le problème posé. On utilisera la notation $d(x_i, x_j)$ pour désigner la distance entre les sommets x_i et x_j . Quelle est la complexité de cet algorithme (en nombre de calcul de distance $d(x_i, x_j)$) ?
4. Comment à la suite de l'algorithme précédent peut-on trouver les deux chemins p_0, p_1, \dots, p_{n_p} et m_0, m_1, \dots, m_{n_m} ? Quelle est la complexité de cette étape ?

²Rappelons que la longitude est l'angle par rapport au méridien de Greenwich. Il s'agit d'une valeur entre -180° et $+180^\circ$ qui détermine la position est-ouest d'un point à la surface de la Terre. La ligne de changement de date correspond à une démarcation dans la zone limite entre -180° et $+180^\circ$ qui correspond à la limite entre les fuseaux horaires $+12h$ et $-12h$.

³la description des chemins est l'objet de la question suivante