École polytechnique — Majeure informatique
Promotion 2000

Cours de conception et analyse d'algorithmes

PC 2 – plus courts chemins et programmation dynamique

1. Triangle de Pascal

On rappelle que le coefficient binomial C_n^k est le nombre de sous-ensembles à k éléments d'un ensemble à n éléments. Proposer un algorithme de calcul de C_n^k basé sur la programmation dynamique.

2. Insertion d'espaces dans un texte

Lorsque l'on souhaite chiffrer un texte (par exemple par substitution lettre par lettre), on commence par en supprimer tous les espaces. Le destinataire, une fois le texte déchiffré, se retrouve donc avec un bloc de texte dans lequel il faut réinsérer des espaces.

On suppose que l'on dispose d'un dictionnaire contenant la liste de tous les mots valides (y compris les pluriels, féminins, formes conjuguées des verbes, etc), accessible par une fonction booléenne *est-dans-dico(mot m)*. Proposer un algorithme basé sur la programmation dynamique pour prendre le bloc de texte donné en entrée et décider s'il existe une insertion d'espaces découpant le bloc en une suite de mots valides. On écrira le pseudo-code correspondant. Quelle est la complexité de l'algorithme ? Le compléter pour que de plus il calcule un découpage lorsqu'il en existe un.

3. Organisation d'un pot

Une PDG souhaite organiser un pot pour fêter la décoration qui vient de lui être décernée. Les relations hiérarchiques dans l'entreprise ont une structure d'arbre dont la PDG est la racine. Elle souhaite qu'un employé ne soit pas invité en même temps que son supérieur immédiat. Qui plus est, à chaque employé est associée une "note de convivialité". Comment prévoir la liste des invités de façon à maximiser la somme des notes de convivialité ? Comment assurer que la PDG sera invitée ?

4. Arbre binaire de recherche optimal

On suppose dans cet exercice que l'on souhaite construire un arbre binaire de recherche, pour lequel on dispose d'informations sur la fréquences de chaque étiquette. (Par exemple ces étiquettes repésentent les mots-clé d'un langage de programmation : begin, do, else, if, then, while, end,...). Proposer un algorithme de type programmation dynamique qui construit l'arbre binaire de recherche qui minimise le nombre moyen de comparaisons à effectuer pour reconnaître un élément.

5. Ordonnancement de tâches

On considère un ensemble de n tâches. La tâche T_i a une durée d'exécution p_i et une date d'échéance d_i.

i. Soit un sous-ensemble des tâches qui soit exécutable sans retard ; rappeler comment on peut trouver un ordonnancement sans retard.

ii. Proposer un algorithme de type programmation dynamique pour déterminer l'ensemble de tâches sans retard de cardinal maximum. Les tâches étant ordonnées par date d'échéance, on considèrera pour cela la durée minimale des ordonnancements des sous-ensembles de tâches inclus dans $\{T_1, \cdots, T_j\}$, exécutables sans retard, et de cardinal supérieur ou égal à w.

iii. Variante : la tâche T_i a de plus une pénalité w_i associée, à payer lorsque les délais ne sont pas respectés. Comment minimiser les pénalités à payer ?

6. Algorithme de Viterbi

On considère un graphe orienté de reconnaissance vocale $G = (S, A)$. Chaque arc est étiqueté avec un phonème tiré d'un ensemble fini. Le graphe modélise une personne parlant un langage restreint. Chaque chemin du graphe partant d'un sommet distingué v_0 correspond à une séquence de sons pouvant être produite par le modèle. L'étiquette d'un chemin est définie comme la concaténation des étiquettes de ses arcs.

i. Proposer un algorithme qui, étant donné une séquence $s = s_1 \dots s_n$ de phonèmes, retourne un chemin de G qui commence en v_0 et d'étiquette s si un tel chemin existe.

ii. On associe maintenant à chaque arc (u, v) une probabilité de transition $p(u, v)$ représentant les chances de passer par l'arc (u, v) à partir du sommet u. La somme des probabilités des arcs sortant d'un sommet vaut 1. La probabilité d'un chemin est le produit des probabilités de ses arcs. Étendre la réponse de la question précédente pour que le chemin retourné soit un chemin *le plus probable* parmi les chemins partant de v_0 et d'étiquette s. Quel est le temps d'exécution de l'algorithme ?

7. Problème du voyageur de commerce

Etant donné un graphe complet non-orienté à n sommets $V = \{1, \dots, n\}$, dont chaque arête a une longueur, le problème du voyageur de commerce consiste à trouver un cycle qui passe une et une seule fois par chacun des sommets du graphe, et de longueur totale minimale. (Ce problème est NP-complet.)

i. Quelle est la complexité de l'algorithme d'énumération exhaustive ?

ii. Soit $C(S, k)$ la longueur minimale d'un chemin pour aller de 1 à k en passant une fois et une seule par tous les sommets de S, où S est un sous-ensemble de V. Exprimer $C(S, k)$ par récurrence. En déduire un algorithme de type programmation dynamique. Quelle est sa complexité ? Comparer.