École polytechnique — Majeure informatique
Promotion 2004

Cours « Conception et analyse d'algorithmes »
TD n° 2 – plus courts chemins et programmation dynamique
24 janvier 2007

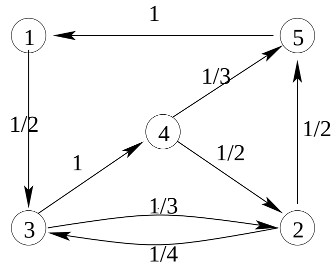
1. Plus courts chemins dans un graphe avec un arc négatif

On considère un graphe orienté $G = (X, A)$, un sommet $x_0 \in X$ et une valuation v des arcs telle qu'il existe un seul arc a_0 de valuation négative.

Donner un algorithme basé sur celui de Dijkstra qui donne les longueurs des plus courts chemins entre x_0 et tous les autres sommets du graphe.

2. Roy-Warshall sur un exemple

On considère le graphe orienté $G = (X, A)$ ci-dessous, à chaque arc duquel est associée une fiabilité (positive ou nulle).



La fiabilité d'un chemin est donnée par le produit des fiabilités des arcs du chemin. Exécuter l'algorithme de Roy-Warshall pour calculer les chemins de fiabilité maximale entre toutes les paires de sommets de G.

3. Algorithme de Viterbi

On peut se servir de la programmation dynamique sur un graphe orienté de reconnaissance vocale $G = (S, A)$. Chaque arc (u, v) est étiqueté avec un phonème $s(u, v)$ tiré d'un ensemble fini P. Ce graphe est un modèle formel d'une personne parlant un langage restreint. Chaque chemin du graphe partant d'un sommet distingué v_0 de S correspond à une séquence de sons pouvant être produite par le modèle. L'étiquette d'un chemin est définie comme étant la concaténation des étiquettes des arcs du chemin.

i. Décrire un algorithme efficace qui, étant donné un graphe G étiqueté, un sommet distingué v_0 et une séquence $s = s_1 \dots s_n$ de caractères de P, 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 $p(u, v)$. 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* partant de v_0 et d'étiquette s. Quel est le temps d'exécution de l'algorithme ?

4. Alignement de familles de séquences

i. Proposer un algorithme en $O(n^3)$ pour trouver le meilleur alignement de 3 séquences de longueur n pour une fonction de distance entre caractères $\delta : (\Sigma \cup \{-\})^2 \to \mathbb{N}$ donnée. δ mesure la distance entre les caractères, ainsi $\delta(\sigma, \sigma') = \delta(\sigma', \sigma) \geqslant 0$ et $\delta(\sigma, \sigma) = 0$.

ii. Proposer un algorithme qui étant donné deux mots u et v et une fonction de similarité δ, détermine en $O(n^2)$ les facteurs u' de u et v' de v de valeur d'alignement maximum. Ici la fonction δ représente la valeur de l'alignement, ainsi $\delta(-, -) = 0$, $\delta(\sigma, \sigma) > 0$, et $\delta(\sigma, \tau) < 0$ pour $\sigma \neq \tau$.

5. Sous-séquences maximales

i. Étant donnée une suite de valeur $X_1, \ldots, X_n$, avec $X_i \in \mathbb{R}$, donner un algorithme qui trouve en temps linéaire i et j tels que $\sum_{k=i}^{j} X_k$ soit maximale.

ii. Généraliser le problème précédent au cas de séquences circulaires.

iii. Généraliser le problème des sous-séquences maximales au cas où les variables X_i sont sur les sommets d'un arbre et où on souhaite trouver le sous-arbre dont la somme des X_i soit maximale.