INF431

Premiers exercices d'algorithmique
Sujet proposé par Bruno Salvy & Gilles Schaeffer
Version: 1475:2400M

1 Variations sur les mariages stables

1.1 Postes multiples

L'algorithme de Gale et Shapley permet de trouver une affectation de n candidats à n services d'une administration, chaque service classant les n candidats selon ses propres critères et chaque candidat classant les n services selon ses préférences. Supposons maintenant qu'il n'y ait que m services avec $m < n$, mais que chaque service dispose de plusieurs postes : plus précisément on suppose que le i-ème service a toujours une unique liste de préférence mais qu'on doive lui affecter n_i candidats (avec $\sum_{i=1}^{m} n_i = n$).

Question 1 Comment résoudre ce nouveau problème d'affectation ? ◇

1.2 Préférences de caste

Supposons que notre administration soit noyautée par les anciens élèves d'une célèbre école (les Y) : chaque service désire avant tout recruter un Y pour bénéficier de son réseau d'amitiés, et classe donc systématiquement les Y devant les autres candidats. Inversement tous les candidats classent les postes de direction devant les postes d'exécutants.

Question 2 Montrer que s'il y a k postes de direction et k candidats qui sont Y, alors chaque poste de direction est occupé par un Y. ◇

1.3 Préférences incomplètes

L'expérience montre qu'il existe des individus qui préfèrent rester célibataire que d'être marié à une personne qui ne leur plaît pas suffisamment... On modélise cela en demandant à chaque homme ou femme de se classer dans sa liste de préférence (éventuellement en dernière position). Ceci amène à étendre la définition de stabilité aux couplages non nécessairement parfaits. Un couplage est instable s'il contient des éléments $h \in H$ et $f \in F$ qui se préfèrent mutuellement à leurs fiancés respectifs dans le couplage, ou s'il contient un élément $x \in H \cup F$ qui préfère rester célibataire qu'épouser son ou sa fiancé(e) dans le couplage. De manière équivalente l'instabilité du couplage revient à l'existence d'une paire $(x, y) \in (H \times F) \cup \{(x, x) \mid x \in H \cup F\}$ telle que $y >_x C(x)$ et $x >_y C(y)$, où $C(x)$ désigne le ou la fiancée de x dans le couplage, ou x lui-même si x n'est pas fiancé.

Question 3 Montrer que dans un couplage stable personne ne peut être marié à un individu qui apparaît plus bas que lui-même dans sa liste de préférence. ◇

Cette propriété implique qu'on peut tronquer la liste de chaque individu à partir de la position où il apparaît : on parle de liste incomplète, et on dit que x apprécie y si y apparaît dans sa liste tronquée.

Question 4 Étendre l'algorithme de Gale et Shapley aux listes incomplètes, et montrer :

(a) que l'algorithme termine

(b) que, si une femme reçoit une proposition d'un homme h alors soit elle n'apprécie pas h, soit elle sera dans la suite fiancée à h ou à un homme au moins aussi bien placé que h dans sa liste.

(c) et que le couplage obtenu est stable. ◇

2 Le problème k-somme

De nombreux problèmes de décision ou d'optimisation sont exprimés comme des problèmes de sommes de sous-ensembles. Le plus classique, *somme de sous-ensembles*, demande, étant donnés un ensemble E de N entiers et un entier $S \in \mathbb{N}$, s'il existe un sous-ensemble de E dont les éléments se somment à S. Ce problème est NP-complet, ce qui signifie que l'on ne connaît pas d'algorithme permettant de le résoudre en un nombre d'opérations croissant polynomialement avec N. Il devient cependant de complexité polynomiale si l'on borne *a priori* la cardinalité du sous-ensemble. Il s'agit alors du problème suivant :

> PROBLÈME k-SOMME : Étant donnés un ensemble E de N entiers et un entier S, déterminer s'il existe $s_1, \ldots, s_k$ distincts dans E tels que $s_1 + \cdots + s_k = S$.

Pour la mesure de la complexité dans ce problème, on fait l'hypothèse que l'addition et la comparaison d'entiers ont un coût unitaire, ce qui est réaliste pour des entiers stockés sur un nombre fixe de mots machines, comme ceux du type `int` de Java ou C.

De nombreuses questions de géométrie algorithmique se ramènent au problème k-somme. Par exemple, l'inclusion d'un polygone dans un autre à translation près se réduit à 4-somme. Plus simplement, étant donnés N points du plan, décider si 3 d'entre eux sont alignés est relié à 3-somme. Par ailleurs, 2-somme est utilisé semble-t-il comme question de test dans certains entretiens d'embauche.

Dans tout cet exercice, les algorithmes demandés sont à présenter sous forme de pseudo-code.

2.1 Approche naïve

Question 5 Proposer un algorithme simple pour résoudre le problème 2-somme avec une complexité quadratique. ◇

Question 6 En généralisant cette méthode, proposer un premier algorithme de complexité polynomiale pour le problème k-somme, pour $k \geq 2$. ◇

2.2 Tri

Question 7 Proposer un algorithme permettant de résoudre le problème 2-somme sur une liste triée en complexité seulement linéaire. ◇

Question 8 En déduire un algorithme de complexité quadratique pour 3-somme (l'entrée n'est pas supposée triée). ◇

Cet algorithme, bien que très simple, est pratiquement le meilleur connu actuellement. Le problème de savoir si N^2 est l'ordre de grandeur d'une borne inférieure à la complexité de 3-somme est ouvert.

2.3 Diviser pour régner

La conception d'algorithmes efficaces repose souvent sur un principe simple : diviser pour régner. Cette idée fera l'objet du Cours 4. En voici quelques conséquences.

Question 9 Déduire de la question 7 un algorithme de complexité $O(N \log N)$ pour 2-somme. (On ne demande pas le pseudo-code.) ◇

Cette complexité $O(N \log N)$ est également une borne inférieure pour ce problème. Notre algorithme est donc optimal !

Question 10 (Plus difficile). Proposer un algorithme de complexité $O(N^2 \log N)$ pour 4-somme. ◇

La même idée montre que l'on peut obtenir un algorithme résolvant k-somme lorsque k est pair en complexité $O(N^{k/2} \log N)$, et une variante en $O(N^{(k+1)/2})$ lorsque k est impair. On est loin de la méthode naïve de la question 6 !

Références

[1] J. Kleinberg et E. Tardös. *Algorithm Design*. Pearson, 2006.

[2] D. E. Knuth. *Mariages stables*. Presses de l'université de Montréal, 1976.

[3] A. E. Roth et J. H. Vande Vate. Random paths to stability in two-sided matching. *Econometrica*, 1990.

[4] Anka Gajentaan et Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry. Theory and Applications*, 5(3):165–185, 1995. ISSN 0925-7721.

[5] Nir Ailon et Bernard Chazelle. Lower bounds for linear degeneracy testing. *Journal of the ACM*, 52(2):157–171, 2005. ISSN 0004-5411.

[6] Jeff Erickson. Lower bounds for linear satisfiability problems. *Chicago Journal of Theoretical Computer Science*, pages Article 8, 28 pp. (electronic), 1999. ISSN 1073-0486.