ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 8 22 November 2010

1. Sorting

a) Trouver un algorithme pour trier exactement 3 éléments et qui utilise en moyenne $\frac{16}{6}$ comparaisons.

b) Supposons que nous avons un tableau de N éléments avec $a[i].key = N - i$ pour tout $i = 0, \ldots, N-1$ (c'est-à-dire que a est trié dans l'ordre inverse, de façon décroissante). Si nous trions ce tableau avec BubbleSort, quel sera le nombre exact d'échanges nécessaires?

c) Supposons que nous avons un tableau de N éléments $a[0], \ldots, a[N-1]$. Soit $(p_0, \ldots, p_{N-1})$ la permutation de $\{0, \ldots, N-1\}$ telle que

$$a[p_0].key \leq \ldots \leq a[p_{N-1}].key.$$

Montrer que si on veut trier a avec BubbleSort, il faudra au moins q passages, où

$$q = \max_{0 \leq i \leq N-1} (p_i - i).$$

2. Nombre moyen d'échanges avec **SelectionSort**

Nous voulons calculer le nombre moyen d'échanges dont a besoin l'algorithme SelectionSort pour trier une suite de longueur N. (Ici *moyen* veut dire la moyenne sur tous les inputs possibles, c'est-à-dire toutes les permutations d'une suite de longueur N.)

a) Supposons que $N = 3$. Nous avons donc une suite de 3 éléments à ordonner. Supposons que les clés valent $0, 1$ et 2. Pour chacune des 6 permutations possibles de ces 3 éléments, combien faut-il à SelectionSort d'échanges pour ordonner la suite?

En supposant que chacune de ces permutations a la même probabilité, quel est le nombre moyen d'échanges nécessaires?

b) Supposons maintenant que N est quelconque. L'algorithme fait $N - 1$ itérations (dans le programme la variable i va de 0 à $N - 2$).

Pendant l'itération i il faudra faire un échange si et seulement si le $i^{\text{ème}}$ élément de la suite n'est pas à la bonne position. On voit aussi qu'au début de l'itération i les éléments $0, \ldots, i-1$ sont tous à la bonne place.

(i) Etant donné une permutation aléatoire de $0, \ldots, k-1$, quelle est la probabilité que 0 soit à la bonne place (c'est-à-dire en premier)?

(ii) Montrer que le nombre moyen d'échanges dont a besoin SelectionSort pour trier une suite à N éléments est égal à

$$\frac{N-1}{N} + \frac{N-2}{N-1} + \ldots + \frac{2}{3} + \frac{1}{2}.$$

***3. Dégénérescence de* QuickSort**

Rappelons que QuickSort nécessite en moyenne $O(N \log N)$ comparaisons si la permutation des clés est aléatoire et uniformément choisie. Néanmoins, il faut à QuickSort $\Omega(N^2)$ comparaisons dans le pire des cas.

Intuitivement, QuickSort aura une meilleure performance si le pivot choisi est plus proche de la médiane de la suite (pour qu'elle soit coupée en deux sous-suites de même taille lors de l'appel récursif). Par contre, si le pivot se trouve parmi les plus grands (ou plus petits) éléments de la suite, il faudra faire plus d'appels récursifs, et donc plus de comparaisons. Le choix du pivot est donc crucial.

a) Si on choisit toujours comme pivot le dernier élément de la suite, montrer que pour tout N on peut trouver une suite de N éléments pour laquelle QuickSort nécessite $\Omega(N^2)$ comparaisons.

b) Considérons la stratégie suivante: on prend comme pivot toujours la médiane des éléments au début, au milieu et à la fin. Puis on échange cet élément avec le dernier élément de la suite (si nécessaire), et on procède ensuite comme dans le cours. (C'est la *3-median strategy.*) Montrer que pour tout N on peut trouver une suite de N éléments pour laquelle QuickSort nécessite $\Omega(N^2)$ comparaisons.