ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Srie d'exercices 8 14 Nov 2011

1. Le codage de Huffman

Lesquels des codes suivants ne sont pas des codes de Huffman (pour aucune fréquence de lettres)?

$$(0, 10, 11), \qquad (00, 01, 10, 110), \qquad (01, 10).$$

2. Codes Huffman ternaires

a) Généraliser l'algorithme de Huffman aux mots de codes ternaires (c'est-à-dire aux mots de code utilisant les symboles 0, 1 et 2).

b) Donner un code de Huffman ternaire pour l'ensemble de fréquences $(\frac{1}{45}, \frac{2}{45}, \frac{3}{45}, \frac{4}{45}, \frac{5}{45}, \frac{6}{45}, \frac{7}{45}, \frac{8}{45}, \frac{9}{45})$

c) Donner un code de Huffman ternaire et binaire pour l'ensemble de fréquences $(\frac{1}{21}, \frac{2}{21}, \frac{3}{21}, \frac{4}{21}, \frac{5}{21}, \frac{6}{21})$

d) Calculer la longueur moyenne dans les deux cas.

3. Shell Sort

a) Utiliser l'algorithme ShellSort avec les incréments $2, 1$ pour ordonner la suite de clés $(4, 3, 2, 6, 1, 5)$. Combien d'échanges ont été utilisés?

b) Montrer que si une suite de longueur N est 2-triée et 3-triée alors on peut la trier en utilisant au plus $N - 1$ mouvements.

4. Nombre moyen d'échanges avec Selection Sort

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 toutes 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 permuation 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

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

5. Tri par échanges adjacents

Les algorithmes de tri élémentaires se réalisent souvent en échangeant uniquement des éléments adjacents (par exemple Bubble sort). Considérons la version alternative de InsertionSort suivante:

Call: InsertionSort(a)
Input: Suite a d'objets avec des clés étant des entiers.
Output: Transformation de a de sorte à avoir $a[i].key \leq a[i+1].key$ pour $0 \leq i < N-1$

for $i = 1, \ldots, N-1$ **do**
  $j \leftarrow i - 1$
  while $a[j].key > a[j+1].key$ et $j \geq 0$ **do**
    $a[j+1] \leftrightarrow a[j]$
    $j = j - 1$

a) Comparer cette version de l'algorithme à la première version donnée dans le cours. Calculer le nombre d'échanges et de comparaisons et comparer ces nombres à la version du cours.

b) Soit σ une permutation aléatoire des éléments $\{1, \ldots, N\}$. Montrer que $\forall i \in \{1, \ldots, N\}$ et $\forall j \in \{0, \ldots, \lfloor N/2 \rfloor\}$ on a:
$$P(|\sigma(i) - i| = j) \geq \frac{1}{N}$$
($|\sigma(i) - i|$ représente la distance entre la position de i dans $(1, \ldots, N)$ et sa nouvelle position dans la permutation $(\sigma(1), \ldots, \sigma(N))$).

c) En déduire que la moyenne sur tous les σ de $|\sigma(i) - i|$ est supérieure ou égale à $\frac{N}{9}$, pour tout $i \in \{1, \ldots, N\}$ fixé, et N assez grand.

d) Conclure du point précédent que tout algorithme de tri qui se limite à échanger des éléments adjacents est $\Omega(N^2)$ en moyenne.