Algorithmique et programmation
Louis Granboulan

1 Tri Shell

1. Rappels sur le tri par insertion directe

   Soit un tableau contenant n valeurs distinctes. Le tri par insertion directe consiste à insérer un par un les éléments de la liste dans la fin du tableau, contenant les éléments déjà triés.

   (a) Pour une permutation σ, on appelle inversion un couple $i < j$ tel que $\sigma(i) > \sigma(j)$. Évaluer le nombre moyen d'inversions d'une permutation de n éléments.

   (b) Étudier la complexité de l'algorithme en nombre de comparaisons et d'affectations.

   (c) Réfléchir aux différentes façons d'améliorer cet algorithme

2. Tri Shell

   L'idée est d'essayer de gagner sur le nombre d'affectations en insérant dans des sous-listes (et donc en faisant faire de plus grands déplacements). L'algorithme consiste en un tri par insertion dans des sous-listes contenant de plus en plus d'éléments, de telle sorte que les éléments soient de plus en plus proches de leur position finale (D. Shell, 1959).
   On dit qu'un tableau est h-trié si les h sous-tableaux regroupant les éléments d'indice différant d'un multiple de h sont triés. C'est-à-dire que les suites $(x_i, x_{i+h}, x_{i+2h}, ...)$ pour $i = 1, ..., h$ sont triées.
   Pour le tri Shell, on choisit une suite entière décroissante $h_t, h_{t-1}, ..., h_1$ avec $h_1 = 1$. On commence par h_t-trier le tableau, puis on continuer avec h_{t-1}, etc. Le h_s-tri se fait par exemple par insertion directe, pour chaque sous-tableau.

   (a) Prouver la correction de cet algorithme de tri.

   (b) Prouver que le h-tri d'un ensemble k-trié donne un ensemble k-trié. En déduire que le cas $h_s = 2^{s-1}$ n'est pas optimal.

   (c) Évaluer le nombre moyen d'inversions d'une permutation 2-triée.

      i. Calculer le nombre de permutations 2-triées.

      ii. Montrer que la somme des nombres d'inversions des permutations 2-triées de n éléments (avec $k = \lfloor n/2 \rfloor$) est $A_n = \sum_{i=0..k,j=0..k} |i-j| \binom{i+j}{j} \binom{n-i-j-1}{k-j}$.

      iii. Prouver $A_{2k+1} = 2A_{2k}$. On peut montrer que $A_{2k} = k4^{k-1}$.

      En déduire la complexité de l'algorithme dans le cas où on fait un 2-tri suivi d'un 1-tri.

   (d) Regarder le cas où on fait un h-tri suivi d'un 1-tri.

   (e) Regarder le cas où $h_s = 2^s - 1$.

Knuth, *The Art of Computer Programming, vol. 3 :* ***Sorting and Searching***, pp. 87–92.

2 Fractions égyptiennes

On appelle **fraction égyptienne** l'inverse d'un nombre entier.

1. Trouvez plusieurs algorithmes différents pour décomposer un rationnel positif en une somme de fractions égyptiennes distinctes.
2. Réfléchir au nombre minimum de termes pour représenter un rationnel et à la minimisation du plus grand dénominateur d'une représentation.

`http://www.ics.uci.edu/~eppstein/numth/egypt/`.