

TD9 – M02 : Complexité

Algorithmes de tri

Michel Van Caneghem

20 novembre 2004

1. **Radix-sorting** : Montrez que ce tri est correct, quand on parcourt les bits de la droite vers la gauche.

2. **Insertion Sort** : Montrez que le nombre de comparaisons et le nombre d'échanges moyen est donné par $N^2/4$ pour une liste aléatoire de taille N .

En fait pour raisonner, il faut compter le nombre d'inversion (c'est à dire le nombre de fois qu'un élément est plus petit qu'un élément qui le précède dans la liste). Il est assez facile de voir que le nombre de comparaison du tri par insertion est égale au nombre d'inversions.

Remarque : une légère modification de cet algorithme, en comparant par rapport au milieu chaque fois (binary insertion-sort) donne une complexité de l'ordre de $O(n \log n)$ dans le meilleur des cas, mais hélas toujours $O(n^2)$ en moyenne.

3. **Un petit tri** Montrez que l'on peut trier une liste de 5 éléments avec seulement 7 comparaisons. Comme $\lceil \log_2 5! \rceil = 7$, on ne peut pas faire mieux. Pourquoi ?

On pourra montrer (faute de mieux), qu'avec merge-sort (ou binary insertion-sort), on arrive à trier avec au plus 8 comparaisons.

(Remarque : Attention : ce n'est pas si facile que cela en a l'air ! !).

4. **Quicksort** :

(a) Montrez que le nombre moyen d'échanges effectués lors du premier partitionnement (avant le croisement des pointeurs) est $(N - 2)/6$.

(b) Montrez que le nombre d'échanges de Quicksort est donné par :

$$(n + 1)(H_{n+1} - 5/2)/3 + 1/2$$

