

TD10 – M02 : Complexité

Complexité de MergeSort et quelques exercices sur les séquences

Michel Van Caneghem

28 novembre 2004

1 MergeSort

Le but est de trouver la formule qui donne la complexité du tri fusion (Merge Sort), c'est à dire le nombre de comparaisons nécessaires pour trier une liste de taille N .

1. Montrez que la formule qui donne le nombre de comparaison de MergeSort est :

$$C_N = C_{\lfloor N/2 \rfloor} + C_{\lceil N/2 \rceil} + N - 1$$

avec $C_1 = 0$.

2. On va supposer que $N = 2^n$. Montrez que dans ce cas on trouve :

$$C_N = N \log_2 N - N + 1$$

Pour un N quelconque, cela est plus compliqué, c'est ce que l'on va voir maintenant.

3. On va d'abord s'intéresser à un problème plus simple : le nombre de comparaisons nécessaires pour une recherche binaire. On peut montrer que la récurrence est donnée par : $B_N = B_{\lfloor N/2 \rfloor} + 1$ avec $B_1 = 1$. Montrez que la solution de cette récurrence est :

$$B_N = \lfloor \log_2 N \rfloor + 1$$

4. Si on considère la récurrence "dérivée" de C_N définie par $D_N = C_{N+1} - C_N$, montrez que l'on trouve :

$$D_N = D_{\lfloor N/2 \rfloor} + 1$$

avec $D_1 = 1$. En déduire la valeur de D_N

5. Montrez alors que l'on peut exprimer C_N sous la forme :

$$C_N = N - 1 + \sum_{k=1}^{N-1} \lfloor \log_2 k \rfloor$$

6. en déduire que :

$$C_N = N \lfloor \log_2(N-1) \rfloor + N + 1 - 2^{\lfloor \log_2(N-1) \rfloor + 1}$$

Remarque On peut se demander si cela vaut le coup d'avoir une formule si complexe. Pour le tester, j'ai pris $N = 1000000$ et j'ai calculé C_N avec la formule établie au point 2, par rapport à celle du point 6. On trouve :

– dans le premier cas : $C_N = 18931600$ comparaisons

– dans le second cas : $C_N = 18951424$ comparaisons

On peut donc conclure que cela ne vaut pas le coup de se compliquer la vie (mais on ne le sait qu'après avoir fait le travail !!). Surtout qu'au point de vue pratique, j'ai trouvé $C_N = 18674193$. Cela veut probablement dire qu'il y a un autre phénomène plus important qui n'a pas été analysé (je vous laisse le soin de découvrir lequel ? et si vous êtes très fort, d'essayer de l'analyser).

2 Les séquences

1. **Un alignement** En utilisant un score de 1 pour une égalité, -1 pour une différence et -2 pour un gap, donnez les meilleurs alignement globaux des 2 séquences suivantes :

AAAG
ACG

2. **Alignement local** Quel est le meilleur alignement local des 2 séquences suivantes :

ATACTACGGAGGG
GAACGTAGGCGTAT

On utilisera les mêmes paramètres que la question précédente.

3. **Combien** Vous avez vu en cours l'algorithme classique de programmation dynamique pour aligner deux séquences d'ADN ou d'acides aminés. On cherche à aligner deux séquences de taille m et n ($m < n$). Exprimer en fonction de n et m le nombre d'alignements distincts possibles. On considèrera les alignements : $\begin{pmatrix} - & x \\ y & - \end{pmatrix}$ et $\begin{pmatrix} x & - \\ - & y \end{pmatrix}$ comme identiques.

Simplifiez la formule pour $m = n$. Combien y-a-t-il d'alignements pour $n = 1000$?

