Calcul de complexité du tri fusion

Supposons d'abord que la taille du tableau initial est $N = 2^p$ une puissance de 2. On a donc alors (complexité en nombre de tests) :

$$C(N) = \begin{cases} 1 & \text{si N=1 (p=0)} \\ 2C(\frac{N}{2}) + N & \text{sinon} \end{cases}$$

(deux appels récursifs avec des tableaux de taille divisée par 2, plus le coût de la fusion).

Si on réécrit :

$$C(2^p) = \begin{cases} 1 & \text{si p=0} \\ 2C(2^{p-1}) + 2^p & \text{sinon} \end{cases}$$

Ou encore, si on divise par 2^p :

$$\frac{C(2^p)}{2^p} = \begin{cases} 1 & \text{si p=0} \\ \frac{C(2^{p-1})}{2^{p-1}} + 1 & \text{sinon} \end{cases}$$

En posant $u_p = \dfrac{C(2^p)}{2^p}$ on a alors :

$$u_p = \begin{cases} 1 & \text{si p =0} \\ u_{p-1} + 1 & \text{sinon} \end{cases}$$

On reconnaît une suite arithmétique, dont on peut calculer le terme général $u_p = 1 + p$, d'où il vient $C(2^p) = 2^p u_p = p.2^p + o(2^p)$. Pour repasser "en N", il suffit de s'apercevoir que $p = log(N)$, et donc on a finalement :

$$C(N) = O(N.log(N))$$

Si maintenant le tableau n'est pas de taille 2^p, il suffit de mettre N en sandwich entre deux puissances de 2 sucessives :

$$2^p < N < 2^{p+1}$$

La fonction de complexité étant croissante, on a donc

$$C(2^p) \leq C(N) \leq C(2^{p+1})$$

soit encore

$$p.2^p \leq C(N) \leq (p+1)2^{p+1}$$

et donc

$$k_1 N log(N) \leq C(N) \leq k_2 log(N)$$

et finalement

$$C(N) = O(N.log(N))$$