

Tri par fusion ou Merge Sort

def tri-fusion(tab):

if len(tab) == 1:

return tab

else:

tabg = tri-fusion tab[0: len(tab)//2]

tabd = tri-fusion tab[len(tab)//2: len(tab)]

return fusion(tabg, tabd)

où fusion(tab-1, tab-2) est une fonction qui fusionne deux tableaux triés tab-1 et tab-2.
en temps

Soit $T(n)$ le temps d'exécution de tri-fusion
sur un tableau de taille n . On a:

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ T(\frac{n}{2}) + T(\frac{n}{2}) + c \frac{n}{2} & \text{si } n > 1 \end{cases}$$

⊗

On a donc

$$\begin{aligned} T(n) &= 2 T\left(\frac{n}{2}\right) + c \frac{n}{2} \\ &\stackrel{(*)}{=} 2 \left[2 T\left(\frac{n}{4}\right) + c \frac{n}{4} \right] + c \frac{n}{2} \\ &= 4 T\left(\frac{n}{4}\right) + 2 \cdot \frac{n}{2} \\ &\stackrel{(*)}{=} 4 \left[2 T\left(\frac{n}{8}\right) + c \frac{n}{8} \right] + 2 \frac{n}{2} \\ &= 8 T\left(\frac{n}{8}\right) + 3 \cdot \frac{n}{2} \\ &\stackrel{(*)}{=} \dots \\ &\stackrel{(*)}{=} 2^k T\left(\frac{n}{2^k}\right) + k \cdot \frac{n}{2} \end{aligned} \quad (1)$$

On sait que $T(1) = 1$, ce qui veut dire que

$$T\left(\frac{n}{2^k}\right) = 1 \text{ lorsque } \frac{n}{2^k} = 1, \text{ i.e., lorsque } n = 2^k, \\ \text{i.e., lorsque } k = \log_2(n). \quad (2)$$

on a

$$\begin{aligned} T(n) &\stackrel{(1)}{=} 2^k T\left(\frac{n}{2^k}\right) + k \cdot \frac{n}{2} \\ &\stackrel{(2)}{=} \underbrace{2^{\log_2(n)}}_{=n} T\left(\underbrace{\frac{n}{2^{\log_2(n)}}}_{=\frac{n}{n}=1}\right) + \log_2(n) \cdot \frac{n}{2} = n + \frac{n}{2} \log_2(n) \\ &\in O(n \log_2(n)) \end{aligned}$$