Borne inférieure Complexité des tris par comparaisons

On veut calculer le nombre minimal de comparaisons d'un algorithme de tri par comparaisons.

Soit un algorithme $\mathcal{A}$ de tri par comparaisons.

On dessine l'arbre binaire des comparaisons pour une entrée à n éléments à trier, il faut pouvoir séparer les $n!$ permutations par l'algo $\mathcal{A}$, autrement dit

$$\#F \geqslant n!$$

(F est l'ensemble des feuilles de l'arbre)

On a les relations suivantes pour un arbre binaire (N est l'ensemble des noeuds et h la hauteur de l'arbre)

$$\#F = \#N + 1 \qquad \text{et} \qquad \#N \leqslant 2^h$$

On a donc

$$n! - 1 \leqslant 2^h$$

Autrement dit

$$h \geqslant \log_2(n! - 1)$$

Or $\log n! = \sum_{k=1}^{n} \log k = \Theta(n \log n)$ [1], donc

$$h \geqslant \Omega(n \log n)$$

.

$$\boxed{\text{Il faut donc au moins } n \log n \text{ comparaisons pour trier } n \text{ élements.}}$$

1. car la moitié des termes sont supérieurs à $\log \frac{n}{2}$, donc on peut minorer la somme par $\frac{n}{2} \log\left(\frac{n}{2}\right)$ et majorer la somme par $n \log n$ donc on a le Θ