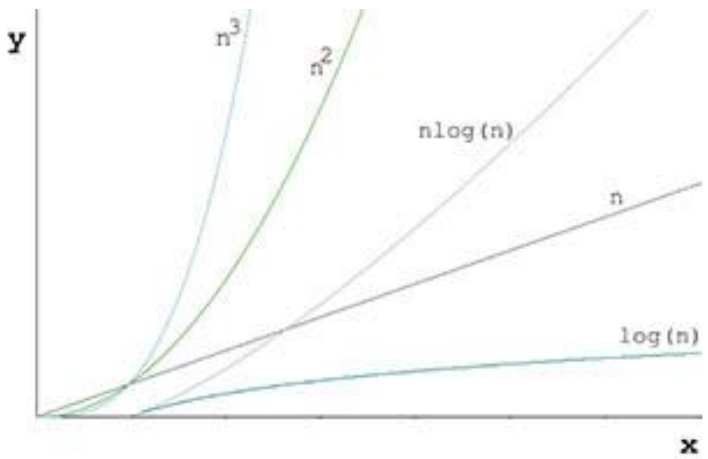
Les principaux types de complexité :

- $\mathcal{O}(1)$, indépendant du contexte (complexité constante)
- $\mathcal{O}(\log(n))$, complexité logarithmique
- $\mathcal{O}(n)$, complexité linéaire
- $\mathcal{O}(n*\log(n))$, complexité quasi-linéaire
- $\mathcal{O}(n^2)$, complexité quadratique
- $\mathcal{O}(n^3)$, complexité cubique
- $\mathcal{O}(2^n)$, complexité exponentielle
- $\mathcal{O}(n!)$, complexité factorielle

Propriétés importantes de l'analyse :

- $f(n) + g(n)$ est en $\mathcal{O}(\max(f(n),g(n))$
- Si $f(n)$ est en $\mathcal{O}(g(n))$ et $g(n)$ est en $\mathcal{O}(h(n))$, alors $f(n)$ est en $\mathcal{O}(h(n))$
- Chaque fonction appartient à la classe $\mathcal{O}$ de la fonction suivante. Par exemple, $n*\log(n)$ est en $\mathcal{O}(n^2)$



$$\sum_{i=1}^{n} 1 = n = \Theta(n)$$

$$\begin{aligned} C_{worst}(n) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 \\ &= \sum_{i=1}^{n-1} (n-i) \\ &= \frac{n(n-1)}{2} \in \Theta(n^2) \end{aligned}$$

Les principales classes de complexité

$O(1)$	temps constant
$O(\log n)$	logarithmique
$O(n)$	linéaire
$O(n \times \log n)$	tris (par échanges)
$O(n^2)$	quadratique, polynomial
$O(2^n)$	exponentiel (problèmes très difficiles)

Exemple : permutation

fonction permutation (S, i, j)

$tmp := S[i]$,	coût c_1
$S[i] := S[j]$,	coût c_2
$S[j] := tmp$,	coût c_3
renvoyer S	coût c_4

Coût total

$$T(n) = c_1 + c_2 + c_3 + c_4 = O(1)$$

Exemple : recherche séquentielle

fonction recherche (x, S, n)

$i := 1$,
tant que $((i < n)$ **et** $(S[i] \neq x))$ **faire** (n fois)
    $i := i + 1$,
renvoyer $(S[i] = x)$

Pire des cas : n fois la boucle

$$T(n) = 1 + \sum_{i=1}^{n} 1 + 1 = O(n)$$