Qu'est-ce qu'un problème informatique ?

besoin d'avoir des entrées pour un problème informatique

Ex : Résolution de toutes les équations du second degré

Existe-t-il une méthode automatique (= algo) qui :
- étant donnés des entiers a, b, c (avec $a \neq 0$) (ce sont les entrées)
- détermine si $ax^2 + bx + c = 0$ a au moins une solution dans $\mathbb{R}$

Problème = questions portant sur des instances (ou entrées)
(pour les problèmes intéressants, l'instance provient d'un ensemble infini)

Si la réponse attendue est "oui" ou "non", c'est un problème de décision

Un algorithme résout un problème

Entrée ⟶ Algo ⟶ Sortie

Un algo s'arrête sur chaque entrée

$\approx \frac{\sqrt{n}}{2}$ tests pour est_premier(n)

$$\frac{\sqrt{n}}{2} = \frac{n^{1/2}}{2} = \frac{\left(2^{\log_2(n)}\right)^{\frac{1}{2}}}{2} = \frac{2^{\frac{\log(n)}{2}}}{2}$$

K = taille de n : $K \approx \log_2(n)$: $2^{K/2}$ ⟹ pas bien (exponentiel)

(visualgo.net) site pour avoir un affichage graphique d'un algo.

$O()$

ce n'est pas le "=" mathématique, ni l'affectation informatique

⚠

$f : \mathbb{N} \to \mathbb{N}$
$g : \mathbb{N} \to \mathbb{N}$ ⟹ $f = O(g)$ et $g = \Omega(f)$ ssi il existe $c > 0$ telle que pour n suffisamment grand $f(n) \leq c\,g(n)$

Complexités à connaître :

$$O(1) < O(\log(n)) < O(n) < O(n\log(n)) < O(n^2) < O(n^3) < \dots < O(2^n) < O(n!)$$

Formules à connaître

- $1+2+3+\dots+n = \frac{n(n+1)}{2} = O(n^2)$
- $1^2+2^2+3^2+\dots+n^2 = n\,n^2 = n^3 = O(n^3)$
- $1^k+2^k+3^k+\dots+n^k = O(n^{k+1})$
- $\log(n!) = \overbrace{\log(1)+\log(2)+\dots+\log(n)}^{n \text{ fois}} = n\log(n)$

  $(\log(ab) = \log(a)+\log(b))$