Definiamo e studiamo le proprietà delle notazioni asintotiche O al fine di valutare e confrontare tra di loro le velocità con cui crescono le funzioni elementari.

Studiamo il comportamento asintotico delle più comuni funzioni attraverso la notazione O (O grande)

Perché vogliamo stimare la velocità di crescita di funzioni che esprimono le quantità di risorse usate da algoritmi, in funzione della taglia dell'input al problema.

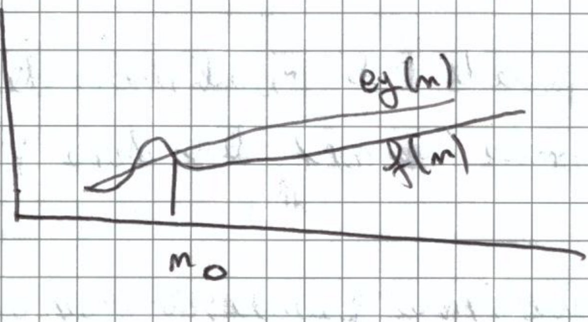
Comportamento asintotico perché in pratica gli algoritmi si usano su input di dimensione molto grande.

Usiamo per valutare i vari algoritmi in base alla velocità.

Date $f: n \in \mathbb{N} \to f(n) \in \mathbb{R}_+$, $g: n \in \mathbb{N} \to g(n) \in \mathbb{R}_+$

$$f(x) = O(g(n)) \iff \exists c>0, \exists n_0 : f(n) \leq g(n), \forall n \geq n_0$$

$f(n) = O(g(n))$ se $f(n)$ non cresce più velocemente di $g(n)$



Esempio

$$10n^3 + 2n^2 + 7 = O(n^3)$$

Proviamo che

$$\exists c, n_0 : 10n^3 + 2n^2 + 7 \leq cn^3, \forall n > n_0$$

Si ha $10n^3 + 2n^2 + 7 \leq 10n^3 + 2n^3 + 7 \leq 10n^3 + 2n^2 + n^3 = 13n^3, \forall n \geq 2$