*Algorithm complexity*

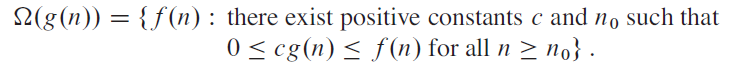
1. *O-Notation*

Descrie comportamentul unui **algoritm in cazul cel mai nefavorabil.**  Este suficient sa specificam o margine superioara pentru timpul de executie



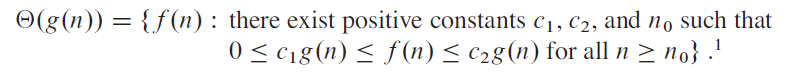
1. *Ω-Notation*

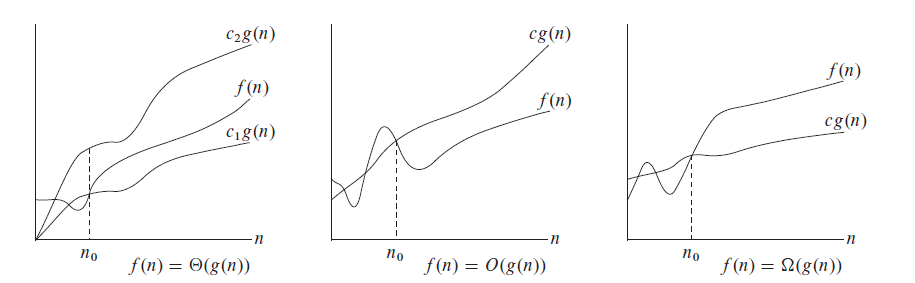
Se foloseste pentru a exprima eficienta algoritmului pornind de la timpul de executie corespunzator **celui mai favorabil caz.** Este suficient sa specificam o margine inferioara pentru timpul de executie



1. *Θ-Notation*

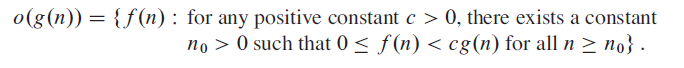
Aceast anotatie decide **daca limitele superioare si inferioare** a unei functii sunt identice sau nu. Daca notatile O si Ω dau acelasi rezultat, atunci notatia Θ are aceasi rata de crestere ca ele. Trebuie sa gasim si margine inferioara si margine superioara.





1. *o-Notation*

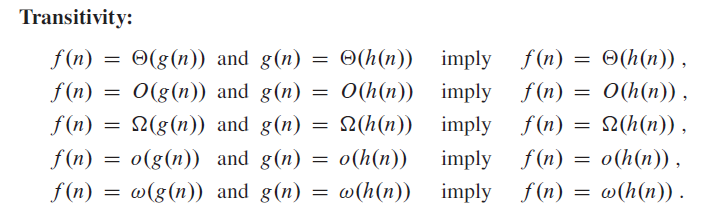
Descrie comportamentul unui **algoritm in cazul cel mai nefavorabil.**  Este suficient sa specificam o margine **strict** superioara pentru timpul de executie



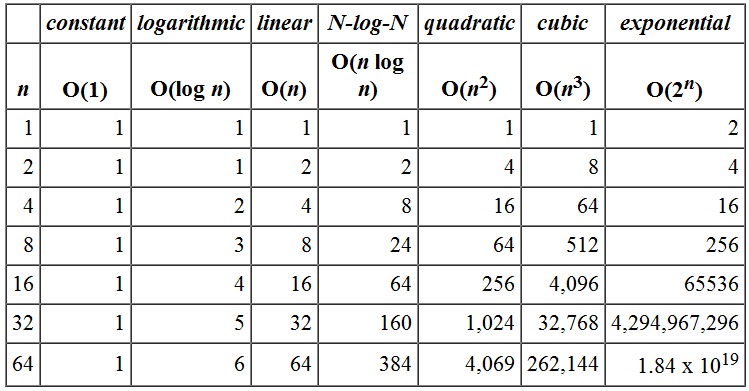
1. *ω-Notation*

Se foloseste pentru a exprima eficienta algoritmului pornind de la timpul de executie corespunzator **celui mai favorabil caz.** Este suficient sa specificam o margine **strict** inferioara pentru timpul de executie





*Cresterea asimptotica a functiilor:*



*Master Theorem*

Metoda Master este o formula ce ajuta la rezolvarea relatiilor recurente de forma T(n) pentru analiza complexitatii de timp in notatia Big O. Teorema este urmatoarea:

