$d(n) = O(f(n)) \Rightarrow a\,d(n) = O(f(n))$ $\forall$ costante $a>0$

Es $\log n = O(n) \Rightarrow 7\log n = O(n)$

$d(n) = O(f(n)),\ e(n) = O(g(n)) \Rightarrow d(n) + e(n) = O(f(n)+g(n))$

Es $\log n = O(n),\ \sqrt{n} = O(n) \Rightarrow \log n + \sqrt{n} = O(n)$

$d(n) = O(f(n)),\ e(n) = O(g(n)) \Rightarrow d(n)e(n) = O(f(n)g(n))$

Es: $\log n = O(\sqrt{n}),\ \sqrt{n} = O(\sqrt{n}) \Rightarrow \sqrt{n}\log n = O(n)$

$d(n) = O(f(n)),\ f(n) = O(g(n)) \Rightarrow d(n) = O(g(n))$

Es: $\log n = O(\sqrt{n}),\ \sqrt{n} = O(n) \Rightarrow \log n = O(n)$

$f(n) = a_d n^d + \dots a_1 n + a_0 \Rightarrow f(n) = O(n^d)$

Es: $5n^7 + 6n^4 + 3n^3 + 100 = O(n^7)$

Tempo lineare $O(n)$: Il tempo di esecuzione dell'algoritmo è al più un fattore costante per la dimensione dell'input

Es: Calcolo del max di n numeri $a_1, a_2 \dots a_n$

```
max ← a1
for i = 2 to n
    if (ai > max)
        max ← ai

return max
```