Soddisfatta per $c = 13$ e $n_0 = 2$

- Possiamo provare che

$$a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 = O(n^k)$$

poiché il polinomio non avrà più elementi di n elevato ma k.

$$a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 \leq |a_k| n^k + |a_{k-1}| n^{k-1} + \dots + |a_1| n + |a_0|$$
$$\leq |a_k| n^k + |a_{k-1}| n^k + \dots + |a_1| n^k + |a_0| n^k = (|a_k| + |a_{k-1}| + \dots + |a_1| + |a_0|) n^k$$
$$= c n^k \quad \Rightarrow a_k n^k + a_{k-1}^{k-1} + \dots + a_1 n + a_0 = O(n^k)$$

- Con $O(n)$ valutiamo la complessità di algoritmi.

Proviamo che

$$\log_2 n = O(n)$$

$$\exists c, n_0 : \log_2 n \leq c n \quad \forall n \geq n_0$$

Per induzione su n: Per $n = 1$ abbiamo $\log_2 1 = 0 \leq 1$

In generale per $n \geq 1$

$$\log_2 (n+1) \leq \log_2 (n+n) = \log_2 (2n) = \log_2 2 + \log_2 n = 1 + \log_2 n \leq 1 + n$$

per ipotesi induttiva

$$\log n \leq n_0 \quad \forall n_0 \geq 1 \Rightarrow \log_2 n = O(n)$$

non

$$\log_a n = (\log_a 2)(\log_2 n) \qquad a \neq 1 \neq b \quad \text{abbiamo}$$

provato che $\log_2 n \leq n$ pertanto

$$\log_a n = (\log_a 2)(\log_2 n) \leq (\log_a 2) n \Rightarrow \log_a n = O(n)$$