

Programmazione dinamica

Monday, 9 October 2023

08:51

- Notazioni asintotiche

- $T(n) \rightarrow$ costo computazionale

Per comprendere quanto cresce utilizzeremo:

- $\theta(g(n)) = \{f(n) \mid \exists c_1, c_2 \exists n_0 \in \mathbb{N} \mid \forall n \geq n_0 \ c_1 g(n) \leq f(n) \leq c_2 g(n)\}$
Sappiamo che la crescita di $f(n)$ è vincolato da come cresce $g(n)$ siccome è compreso da un certo punto in avanti
- $O(g(n))$ è lo stesso di sopra però diciamo che è sempre su
- $\Omega(g(n))$ è lo stesso di sopra però diciamo che è sempre in basso

- $n \in \mathbb{N} \rightarrow$ dimensione input

- Nella programmazione dinamica noi vogliamo trasformare un problema ricorsivo in un problema iterativo

- Per non uccidere lo stack
- Avvengono ottimizzazioni che ci riducono O

- Ci sono 2 modi per trasformarlo:

- Srotolamento

Noi cerchiamo di trovare una soluzione dalla chiamata ricorsiva

"srotolandola"

Es: Risolvere ricorsivamente srotolando

- $f(n) = n^2$
 $f(n-1) = f(n-1)^2 = n^2 - 2n + 1$
Quindi
 $f(n) = f(n-1) + 2n - 1$
Non sto a spiegarlo troppo, siccome non è in esame e non viene utilizzato

Ma in parole povere cerchiamo di srotolare fino a che non troviamo il caso base

- Strategia bottom-up

- Individuiamo i sottoproblemi
Noi risolviamo i problemi più piccoli, che sono banali, e piano piano andiamo a risolvere i problemi più complessi utilizzando il risultato dei risultati precedenti
- Troviamo le equazioni di ricorrenza
- Memorizza in memoria i risultati
- Si applica il bottom-up