

DP => Forza bruta "controllata"

DP => Ricorsione + riuso

DP => Ricorsione + Memoization + **Guessing**

DP => Ricerca di un minimo percorso nel DAG dei sottoproblemi.

Tempo di esecuzione = #Sottoproblemi * Tempo/Sottoproblema

5 Passi

1. Definizione dei sottoproblemi #Contare i sottoproblemi
2. Guessing # Possibili Scelte
3. Collegare le soluzioni dei sottoproblemi # Tempo/Sottoproblema
 - a. Scrivere la ricorrenza
4. Ricorsione + Memoization (oppure Bottom-up (Tabella))
 - a. Check: grafo dei sottoproblemi deve essere aciclico
5. Risolvere problema originario # Extra Time

	Fibonacci	Shortest Paths $\delta(s, v)$, per ogni v
Sottoproblemi (#sottoproblemi)	F_k con $k = 1 \dots n$ n	$\Delta_k(s, v)$ per ogni v in V , per $k = 0, \dots, V -1$ V^2
Guessing (#Possibili scelte)	No 1	#archi entranti in v $\text{indegree}(v) + 1$
Ricorrenza	$F_k = F_{k-1} + F_{k-2}$	$\Delta_k(s, v) = \min\{\delta_{k-1}(s, u) + w(u, v) \mid (u, v) \in E\}$
Time/sottoproblema	$\Theta(1)$	$\Theta(\text{indegree}(v) + 1)$
Check DAG (Time)	OK $\Theta(n)$	OK $\Theta(V) + \Theta(V^2) \sim \Theta(V)$
Problema originario (Extra time)	F_n $\Theta(1)$	$\Delta_{(V -1)}(s, v)$, per ogni v in V $\Theta(V)$

Text Justification

Gisutificare testo Algoritmi e strutture dati llsfgdsgsdfgsdjdlskgjfdlsjgfd Gisutificare testo Algoritmi e strutture dati llsfgdsgsdfgsdjdlskgjfdlsjgfd Gisutificare testo Algoritmi e strutture dati llsfgdsgsdfgsdjdlskgjfdlsjgfd Gisutificare te Algoritmi strutture dati llsfgdsgsdfgsdjdlskgjfdlsjgfd Gisutificare testo Algoritmi e strutture dati llsfgdsgsdfgsdjdlskgjfdlsjgfd

Lista di parole $w[i:j]$

$\text{Badness}(i, j) = \text{infinito se } \text{Length} > \text{PageWidth}$
 $(\text{PageWidth} - \text{Length})^3$

Obiettivo: Suddividere le parole in linee in modo tale da Minimizzare la somma di Badness

N parole => N bit

10000001010100010101010101 => 2^n

1 Definizione del sottoproblema

DP(i) è la Badness per il suffisso Words[i :]

#Sottoproblemi: n

2. Guessing corrisponde a dove iniziare la prossima linea

#scelte = $n - i = O(n)$

3 Ricorrenza

$DP(i) = \min(\text{Badness}(i,j) + DP(j) \text{ for } j = (i+1) \dots (n+1))$

$DP(n) = 0$

Tempo per sottoproblema: **Theta(n)**

4. Il garfo è Aciclico

Tempo totale: #sottoproblemi * tempo/sottoproblema = $\Theta(n^2)$

5. Soluzione Finale

DP(0) è soluzione finale , non c'è **extra time**.