Esercitazione 8 – Programmazione dinamica

Consideriamo nuovamente il problema di equilibrare una sequenza di N frecce adiacenti (esercitazione 6).

►◄►►►◄

Assumiamo N pari

Definiamo uno scontro come due frecce consecutive che puntano una verso l’altra: ►◄

Dato uno scontro, definiamo la parte sinistra come la sequenza di frecce contigue orientate verso destra di lunghezza massima

Analogamente, definiamo la parte destra come la sequenza di frecce contigue orientate verso sinistra di lunghezza massima

Programmazione dinamica

Definiamo uno scontro in equilibrio se la sua parte destra e la sua parte sinistra hanno la stessa lunghezza

Definiamo una sequenza equilibrata se tutti gli scontri che contiene sono equilibrati

Vogliamo determinare il numero minimo di frecce da girare per rendere equilibrata una sequenza

Programmazione dinamica

Soluzione ricorsiva con modello combinatorio (esercitazione 6): tra tutti i possibili sottoinsiemi di frecce da girare, vogliamo selezionare un sottoinsieme che soddisfa le seguenti proprietà:

- Rende la sequenza equilibrata
- Ha cardinalità minima

Modello: powerset – generiamo tutti i possibili sottoinsiemi contenenti frecce da girare

- Disposizioni ripetute: ad ogni passo decido se la freccia va girata o meno (equivalente a generare numeri binari)
- Combinazioni semplici: per ogni cardinalità k seleziono tutti gli insiemi di k frecce da girare

Programmazione dinamica

Soluzione ricorsiva alternativa:

Definiamo $f(a,b)$ il costo (numero di frecce da girare) per riequilibrare la sequenza con indici nell'intervallo

$$[a,b)$$

con b escluso (cfr. libro esercizi per la versione con b incluso)

Una soluzione ottima ha costo $f(0,N)$

Programmazione dinamica

Consideriamo la sottosequenza $[a, b)$. Se la soluzione ottima per la sottosequenza contiene un singolo scontro, $f(a, b)$ corrisponderà al numero di frecce da girare per ottenere che le prime $\frac{b-a}{2}$ frecce siano orientate verso destra, mentre le restanti $\frac{b-a}{2}$ siano orientate verso sinistra

Altrimenti, possiamo decomporre la sequenza $[a, b)$ in due sottosequenze di lunghezza ≥ 2

$$[a, k) \ , \quad [k, b)$$

con $k \in [a + 2, b - 2]$

Programmazione dinamica

Possiamo esprimere $f(a,b)$ come

$$f(a,b) = f(a,k) + f(k,b)$$

e $f(a,k)$ e $f(k,b)$ dovranno essere a loro volta ottime.

In generale, possiamo esprimere $f(a,b)$ come

$$f(a,b) = \min\left[C(a,b), \min_{a+2\leq k\leq b-2}(f(a,k)+f(k,b))\right]$$

dove $C(a,b)$ è il costo di riequilibrio della sottosequenza $[a,b)$ assumendo che contenga un singolo scontro

Programmazione dinamica

Prima soluzione: ricorsione

Seconda soluzione: ricorsione + memoization (top–down)

Terza soluzione: Programmazione dinamica (bottom–up)