

Contents

1	ALGORITMI	1
1.1	Programmazione dinamica	1
1.1.1	Introduzione	1
1.1.2	Approccio al problema	1
1.1.3	Algoritmi visti in classe	1

1 ALGORITMI

1.1 Programmazione dinamica

1.1.1 Introduzione

L'idea è quella di non dover ripetere calcoli già fatti in precedenza, cosa che succede per molto di frequente in un albero di chiamate ricorsive. Una soluzione è quella di "salvare" soluzioni a problemi intermedi in una struttura di supporto è di controllare se un particolare calcolo è già stato fatto. I problemi risolti dalla programmazione dinamica hanno la caratteristica di essere facilmente risolvibili per "piccole" istanze del problema, praticamente l'analogo del caso base in una soluzione ricorsiva. Combinando soluzioni intermedie più semplici da risolvere si ottiene la soluzione dell'istanza completa (e più complessa) del problema originale.

1.1.2 Approccio al problema

Cominciare definendo l'istanza del problema e la soluzione del problema in maniera formale. Può essere utile, ma non è obbligatorio, definire l'equazione di ricorrenza del problema. Anche se in seguito si utilizzerà un approccio di programmazione dinamica questo passaggio ci permette di individuare il caso base e lo step per ipoteticamente raggiungere la soluzione.

Una volta individuata la struttura di supporto più appropriata per il problema definire cosa la struttura contiene ad una particolare posizione. Generalmente contiene la soluzione fino a il/i carattere/i della sequenza/sequenze corrispondente alla posizione della struttura di supporto. Molto importante è definire se la soluzione include o meno il carattere.

1.1.3 Algoritmi visti in classe

1. Longest increasing subsequence (LIS)

Avendo una sequenza \mathbf{X} di lunghezza \mathbf{n} con indice $\mathbf{I} = \{1, \dots, \mathbf{n}\}$, una sottosequenza \mathbf{Z} di \mathbf{X} viene ritenuta valida se gli indici degli elementi di \mathbf{X} che la compongono rispettano un ordine strettamente crescente.

Esempio:

$\mathbf{X} = \langle a, b, c, d \rangle$ $\mathbf{Z} = \langle a, c, d \rangle$ è valida

$\mathbf{X} = \langle a, b, c, d \rangle$ $\mathbf{Z} = \langle d, a, b, c \rangle$ non è valida

Vogliamo trovare la più lunga sottosequenza dove ogni elemento è maggiore del precedente (LIS).

L' algoritmo visto in classe si limita a trovare la lunghezza della LIS.

Istanza del problema:

Una sequenza \mathbf{X} di elementi, la lunghezza di \mathbf{X} è \mathbf{n}

Soluzione del problema:

- Se voglio la LIS: Una permutazione di \mathbf{Z} di \mathbf{X} tale che:

(a) Tra tutte le permutazioni \mathbf{P}