



Progetto “Longest Increasing Subsequence” – Parte II

28 Aprile 2017

1. Programmazione Dinamica bottom-up

Il contesto è ancora quello relativo al problema della sottosequenza crescente più lunga. Nella prima parte hai trasformato un programma per calcolarne la lunghezza (`llis`) applicando la tecnica di *memoization*. A questo punto è facile impostare una nuova soluzione in cui i valori da assegnare alle componenti della matrice (struttura di supporto) vengono calcolati “bottom-up” e l’ordine di elaborazione è controllato direttamente attraverso costrutti iterativi anziché essere delegato ai meccanismi che implementano la ricorsione.

Scrivi un programma funzionalmente equivalente ai precedenti che applica una tecnica bottom-up di *programmazione dinamica* per migliorare ulteriormente le prestazioni. Verifica quindi che i risultati ottenuti siano coerenti con i valori calcolati negli altri modi.

2. Determinazione di una sottosequenza crescente più lunga

Prendendo come modello di riferimento la procedura per calcolare la sottosequenza comune più lunga (*LCS*) a partire dalle lunghezze delle soluzioni per le diverse coppie di postfissi, sviluppata a lezione, formalizza i criteri per individuare un percorso attraverso la matrice che consenta di ricostruire la/una sottosequenza crescente più lunga.

Scrivi quindi un metodo statico `lis` che, dopo aver costruito e assegnato i valori alle componenti della matrice esattamente come già fatto in relazione al punto precedente, completa l’elaborazione ricostruendo la sottosequenza cercata e restituendola nella forma di `Vector<Integer>`. L’istestazione è dunque la seguente:

```
public static Vector<Integer> lis( int[] s )
```

Esempi

```
lis( new int[] {5, 4, 3, 2, 1} )           → [3]
lis( new int[] {47, 38, 39, 25, 44} )      → [38, 39, 44]
lis( new int[] {27, 90, 7, 29, 49, 8, 53, 1, 28, 6} ) → [27, 29, 49, 53]
lis( new int[] {9, 46, 54, 71, 60, 47, 0, 32, 25, 61} ) → [9, 46, 54, 60, 61]
lis( new int[] {54, 52, 42, 33, 14, 40, 37, 61, 53, 1} ) → [33, 40, 61]
```