Laboratorio di Programmazione a.a. 2020-21

Progetto "Longest Increasing Subsequence" – III

12 Maggio 2021

Premessa

Data una sequenza *s* di *n* interi positivi, rappresentata da un array, il seguente programma in Java calcola la lunghezza della più lunga sottosequenza di *s* strettamente crescente (`llis`: *length of the longest increasing subsequence*).

Per una descrizione del problema, dell'algoritmo ricorsivo e dell'applicazione di una tecnica di programmazione dinamica per risolverlo con maggiore efficienza, fai riferimento alle due esercitazioni precedenti (28/04 e 5/05/2021).

```
public static int llis( int[] s ) {   // s[i] > 0 per i in [0,n-1], dove n = s.length
  return llisRec( s, 0, 0 );
}

private static int llisRec( int[] s, int i, int t ) {
  if ( i == s.length ) {              // i = n :  coda di s vuota
    return 0;
  } else if ( s[i] <= t ) {           // x = s[i] ≤ t :  x non può essere scelto
    return llisRec( s, i+1, t );
  } else {                            // x > t :  x può essere scelto o meno
    return Math.max( 1+llisRec(s,i+1,s[i]), llisRec(s,i+1,t) );
  }
}
```

1. Sottosequenza *decrescente* più lunga

Questo problema si formula in termini analoghi a quello della sottosequenza crescente più lunga, salvo che, come indicato dalla denominazione, gli elementi considerati devono costituire una sottosequenza strettamente *de*crescente, anziché crescente, nel rispetto dell'ordine in cui compaiono in *s*.

Modifica e/o integra il programma riportato sopra e scrivi un corrispondente programma *ricorsivo* per determinare la lunghezza della sottosequenza decrescente più lunga (`llds`).

2. Applicazione della tecnica di programmazione dinamica bottom-up

Riutilizza il codice del programma che risolve il problema proposto al punto 2 dell'esercitazione del 5/05/2021 (Progetto "Longest Increasing Subsequence" – II), apportandovi le opportune modifiche e/o integrazioni, per determinare una sottosequenza decrescente più lunga (`lds`). Il programma deve avere una struttura analoga a quella impostata nell'esercitazione precedente, suddivisa in una prima fase per registrare in una matrice i risultati delle ricorsioni relative al programma originale, seguita da una seconda fase che consente di ricostruire una soluzione attraverso un percorso fra gli elementi della matrice.

3. Caricamento del codice prodotto attraverso la piattaforma *e-learning*

Accedi con le tue credenziali alla sezione dedicata al laboratorio del corso di Programmazione attraverso la piattaforma *uniud e-learning* (https://elearning.uniud.it) e seguendo lo schema indicato nel *quiz moodl*e predisposto per questa esercitazione, inserisci i programmi realizzati per risolvere i problemi proposti nei due punti precedenti.

Le soluzioni corrette caricate entro la scadenza del quiz moodle (vedi termini indicati per il quiz stesso) saranno considerati elementi positivi ai fini della valutazione della prova di Laboratorio di Programmazione.

Inoltre, nel caso volessi migliorare le soluzioni relative al progetto "Longest Increasing Subsequence" che hai fornito in precedenza (parti I e II), al termine dell'esercitazione avrai l'opportunità di caricarne una revisione.