Mega 1 parziale

Monday, 20 November 2023 15:02

Nota: l'ho iniziato solamente oggi
Questo file verrà aggiornato prima della fir
Con tutte le soluzioni del emga

1)

Si considerino due sequenze di numeri interi X e Y, di lunghezza n e m rispettivamente, ed una funzione $\phi : Z \rightarrow \{R,N,B\}$ che associa ad ogni numero intero il colore *Rosso* (*R*), *Nero* (*N*) o *Blu* (*B*). Si vuole calcolare, mediante la tecnica della programmazione dinamica, la lunghezza di una più lunga sottosequenza DECRESCENTE comune a X e Y, in cui non compaiano mai due numeri consecutivi a cui e' associato lo stesso colore. RISPONDERE PER PUNTI alle seguenti richieste:

1) esplicitare e definire le variabili che servono per risolvere il problema
2) scrivere l'equazione di ricorrenza per il CASO BASE, giustificando perchè e' fatta in quel modo
3) scrivere la/le equazione/i di ricorrenza per il PASSO RICORSIVO, giustificando perchè e'/sono fatta/e in quel modo
4) scrivere qual è la soluzione del problema, espressa rispetto alle variabili introdotte
5) Scrivere quindi, in pseudocodice, l'algoritmo relativo

1) $X = \{x_0, x_1, \dots, x_n\}$
$x_i \in X \;\forall\, i \leq n$
$Y = \{y_0, y_1, \dots, y_m\}$
$y_j \in Y \;\forall j \leq m$
$\phi : Z \rightarrow \{R, N, B\}$
$S = \{s_1, s_2, \dots, s_k\} = max$
$\forall s \in S,$
$s_1 < s_2 < \cdots < s_k$
$X \cap Y \cap S = S$
$col(s_i) \neq col(s_{i+1})$

Notiamo che questa è una fusione tra LCS ed il LIS, aka chiamata LICS
- Trovare la più lunga sottosequenza
- Tra 2 sequenze

Quindi, avendo notato questo sappiamo che abbiamo bisogno della funzione au
- La prima funzione servirà a trovare il massimo delle sottosequenze
  Chiameremo questa $LICS$
- La seconda funzione servirà a trovare il massimo delle sottosequenze con

Chiameremo $LICS_{aux}$

2) Casi base e passi ricorsivi
   - LICS
     - $i = 0 \vee j = 0$
       $c_{ij} = 0$
       Se i=0 oppure j=0 allora o X oppure Y sono vuoti
       E quindi nulla può essere uguale tra di loro
     - Else
       $c_{ij} = \max\left(c_{ij}^{aux}, i < n, j < m\right)$
       Siccome per trovare una più lunga sottosequenza decrescente no
       sottosequenza più lunga da 1 o i, noi dobbiamo iterare per tutte l
       sottosequenza più lunga
   - $LICS_{aux}$
     - $x_i \neq y_j$
       $c_{ij}^{aux} = 0$
       Caso base, se sono diversi siamo sicuri che il risultato è = 0
     - $x_i = y_j$
       $c_{ij}^{aux} = \max\{c_{nm}^{aux} + 1, n < i, m < j, x_n < x_i, col(x_i) \neq col(x_n)\}$
       Dobbiamo iterare per tutti i valori prima di i e j
       E di tutti questi valori controllare solo quando il valore è decresc
       sono diversi, e quando lo abbiamo trovato, rifacciamo lo stesso co
       metterlo a confronto con quelli successivi

4) Una volta aver trovato tutti gli c_{ij}^{aux}
   Basta prendere il massimo
   Ed esso sarà la nostra solzuione
   $S = \max\left\{c_{ij}^{aux}\right\}$

5) Ora scrivere lo pseudocodice

$$LICS(i,j) = \begin{cases} 0 & i = 0 \vee j = 0 \\ \max(LICSaux(i,j), i < n, j < m) & \end{cases}$$

$$LICSaux(i,j) = \begin{cases} 0 \\ \max\left(LICSaux(n,m) + 1, n < i, m < j, x_n < x_i, col(x_i) \neq co\right. \end{cases}$$

```
LICS(i, j):
    If i=0 or j=0 then
        Return 0
    Max = 0
    For n=0 to i:
        For m=0 to i:
            TempMax = LICSaux(n, m)
            If TempMax > Max:
                Max = TempMax
```

```
                        Max = TempMax
    Return Max

LICSaux(i, j):
    S[] = [i, j]

    For n=1 to i do:
        For m=1 to i do:
            If xn != ym then:
                S[n, m] = 0
            Else:
                tempMax = 0
                For h=1 to n-1 do:
                    For k=1 to m-1 do:
                        If tempMax < S[h, k] and xn < xh and col(
                            Tempmax = S[h, k]
                S[n, m] = Tempmax + 1
    Return S[i, j]
```