

Massima copertura

Dati n segmenti della retta delle ascisse, dove l' i -esimo segmento inizia nella coordinata $a[i]$ e termina nella coordinata $b[i]$, scrivere un algoritmo che trovi il sottoinsieme di segmenti che coprono la maggior parte della retta e non sono sovrapposti. Valutare il costo computazionale dell'algoritmo proposto.

Quadrato binario

Sia $A[1 \dots n, 1 \dots n]$ una matrice di valori booleani 0/1. Scrivere un algoritmo che restituisce la dimensione del più grande quadrato composto da valori 1. Ad esempio, nella matrice seguente, i quadrati di dimensione massima (ve ne sono due, di cui uno evidenziato in rosso) hanno dimensione pari a 4.

1	0	1	0	1	0	0
1	0	1	1	1	1	0
0	1	1	1	1	1	0
0	0	1	1	1	1	0
1	1	1	1	1	1	0
1	1	1	1	1	1	0

I Promessi Sposi

"Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a restringersi, e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia costiera dall'altra parte; e il ponte, che ivi congiunge le due rive, par che renda ancor più sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda ricomincia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni."

Quante volte questo testo contiene la sottosequenza "lucia"?

I Promessi Sposi

"Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti, tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto, a ristringersi, e a prender corso e figura di fiume, tra un promontorio a destra, e un'ampia costiera dall'altra parte; e il ponte, che ivi congiunge le due rive, par che renda ancor più sensibile all'occhio questa trasformazione, e segni il punto in cui il lago cessa, e l'Adda rincomincia, per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo, lascian l'acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni."

Quante volte questo testo contiene la sottosequenza "lucia"?

Alcune considerazioni:

- Due sottosequenze sono diverse (e quindi vanno contate separatamente) se esiste almeno una differenza negli insiemi di caratteri utilizzati.
- Esempio: "did you go" contiene due volte la sottosequenza "dog"....

Vettori ordinati

Si scriva un algoritmo che preso in input n e k , restituisca il numero totale di vettori distinti di lunghezza n , contenenti valori interi compresi fra 1 e k , ordinati dalla relazione \leq . Si discuta la correttezza e la complessità dell'algoritmo proposto.

Ad esempio, dati $n = 4$ e $k = 3$, questi sono i possibili vettori ordinati:

[1, 1, 1, 1], [1, 1, 1, 2], [1, 1, 1, 3], [1, 1, 2, 2], [1, 1, 2, 3],
[1, 1, 3, 3], [1, 2, 2, 2], [1, 2, 2, 3], [1, 2, 3, 3], [1, 3, 3, 3],
[2, 2, 2, 2], [2, 2, 2, 3], [2, 2, 3, 3], [2, 3, 3, 3], [3, 3, 3, 3]

e quindi il valore da restituire è 15.

Spoiler alert!

Massima copertura

Questo problema è molto simile al problema dell'insieme indipendente di intervalli pesati visto a lezione, dove il peso $w[i]$ è pari a $b[i] - a[i]$. Si risolve quindi con una singola chiamata a quella soluzione, in un tempo pari a $\Theta(n \log n)$.

```
SET segmentcover(integer[] a, integer[] b, integer n)
```

```
integer[] w = new integer[1 .. n]
```

```
for  $i = 1$  to  $n$  do
```

```
     $w[i] = b[i] - a[i]$ 
```

```
return maxinterval( $a, b, w, n$ )
```

Quadrato binario

$DP[i, j]$ contiene la dimensione del più grande quadrato composto da soli 1 il cui angolo in basso a destra sia nella posizione (i, j)

$$DP[i, j] = \begin{cases} 0 & A[i, j] = \mathbf{false} \\ 1 & A[i, j] = \mathbf{true} \wedge i = 1 \vee j = 1 \\ \min\{DP[i-1, j], \\ DP[i-1, j-1], \\ DP[i, j-1]\} + 1 & \text{altrimenti} \end{cases}$$

I Promessi Sposi

Input: Testo $T[1 \dots n]$, pattern $P[1 \dots m]$

Sia $D[i, j]$ il numero di occorrenze del prefisso j -esimo del pattern $P(j)$ come sottosequenza del prefisso i -esimo del testo $T(i)$.

$$D[i, j] = \begin{cases} 0 & i = 0 \wedge j > 0 \\ 1 & j = 0 \\ D[i - 1, j] + D[i - 1, j - 1] & i > 0 \wedge j > 0 \wedge T[i] = P[j] \\ D[i - 1, j] & \text{altrimenti} \end{cases}$$

I Promessi Sposi

Utilizziamo un vettore $D[0 \dots m]$ invece che una matrice, in quanto il valore si ottiene a partire dalla sola riga precedente.

```
lucia(ITEM[] T, ITEM[] P, integer n, integer m)
```

```
integer[] D = new integer[0...n]
```

```
integer[] D' = new integer[0...n]
```

```
D[0] = 1
```

```
for j = 1 to m do D[j] = 0
```

```
for i = 1 to n do
```

```
    for j = 0 to m do D'[j] = D[j]
```

```
    for j = 1 to m do
```

```
        if T[i] == T[j] then
```

```
            D[j] = D'[j] + D'[j - 1]
```

```
        else
```

```
            D[j] = D'[j]
```

```
return D[m]
```

5 Maggio e Promessi Sposi

----- Forwarded Message -----

Subject: Sottosequenza Promessi sposi

Date: Fri, 11 Dec 2015 00:25:41 +0100

To: Alberto Montresor <alberto.montresor@unitn.it>

Il 5 maggio nei promessi sposi senza considerare spazi, punteggiatura e numeri, ma considerando gli accenti ci sta:

21975465301516630979573617593825769513857583563025262379789778337947615817191757
43398428321975621542396623347442197637158184228160098596758725678010178001365659
73566045203119340799723612777962220975263078675519750712637479432237655210391918
48601874737423942438531018213728179566210700422537584195776715536664949343794694
74341304486367721199869205484517178136400988317581077715393614892844560303556628
5775372295765872497041607413767080756185495931470763581205734844533068267511860
81613043221797286605904378408112853795888506693820006728695515750235630153301285
93082577269020619288952011873970086359263850877345042074027243309950696549344467
34109698508036913355586284550592994928187284736568396263368466837671143105426910
99608601301418040383501823489133955578653269527834272234364741431604038516826654
82045722458282856692545688317000656065623166181081105288235575975572352074726528
75237693750708795898738364470324968401496146509587976160485483512050002468507459
5216118769351618590697862390987987264814086004487458833092023307

Allego lo script in python che ha generato il risultato (con testo completo dei promessi sposi e del 5 maggio). Ovviamente ha tempi assurdi per svariati motivi.

Vettori ordinati

Sia $C[n, k]$ il numero di vettori ordinati di lunghezza n , contenenti k valori distinti (compresi fra 1 e k). $C[n, k]$ può essere calcolato in maniera ricorsiva come segue:

$$C[n, k] = \begin{cases} 1 & n = 0 \\ \sum_{i=1}^k C(n-1, i) & n > 0 \end{cases}$$

Vettori ordinati

permutazioni-ordinate(integer n , integer k , integer[][] C)

if $n == 0$ **then**

return 1

if $C[n, k] == \perp$ **then**

$C[n, k] = 0$

for $i = 1$ **to** k **do**

$C[n, k] = C[n, k] - \text{permutazioni-ordinate}(n - 1, i, C)$

return $C[n, k]$

Ovviamente, questo richiede una tabella $O(nk)$, per calcolare ogni elemento delle quale saranno necessarie $O(k)$ operazioni, per un costo totale di $O(nk^2)$.

Vettori ordinati

Una soluzione alternativa, più efficiente, calcola $C[n, k]$ nel modo seguente:

$$C[n, k] = \begin{cases} k & n = 1 \\ 0 & k = 0 \\ C[n - 1, k] + C[n, k - 1] & \text{altrimenti} \end{cases}$$

Vettori ordinati

```
permutazioni-ordinate(integer  $n$ , integer  $k$ , integer[][]  $C$ )
```

```
if  $n == 1$  then  
   $\perp$  return  $k$   
if  $k == 0$  then  
   $\perp$  return 0  
if  $C[n, k] == \perp$  then  
   $C[n, k] =$   
    permutazioni-ordinate( $n - 1, k, C$ ) + permutazioni-ordinate( $k, n - 1, C$ )  
return  $C[n, k]$ 
```

Ovviamente, questo richiede una tabella $O(nk)$, per calcolare ogni elemento delle quale saranno necessarie $O(1)$ operazioni, per un costo totale di $O(nk)$.