

Sequenza specchiata

Dato un vettore V di n interi appartenenti all'insieme $\{0, 1\}$, si scriva un algoritmo che restituisca la lunghezza del più lungo sottovettore contiguo formato da k valori 0 seguiti da k valori 1.

Si noti che è possibile che esistano altri valori 0 prima del sottovettore così individuato, oppure altri valori 1 dopo il sottovettore, ma non è possibile che si verifichino entrambe le estensioni, altrimenti il sottovettore non sarebbe massimale. Discutere correttezza e complessità dell'algoritmo proposto.

Esempio: per l'input 00111111100011110000, l'algoritmo deve restituire 6.

Aggiornamento del flusso massimo

Sia $G = (V, E, s, t, c)$ una rete di flusso con valori interi. Si supponga di conoscere un flusso massimo in G , e si supponga che la capacità di un singolo arco $(u, v) \in E$ sia aumentata di una unità. Progettare un algoritmo per aggiornare il flusso massimo in tempo $O(n + m)$ oppure $O(n^2)$

Ballo di fine anno

Una scuola vuole organizzare un ballo di fine anno. Ci sono n maschi e m femmine. Ogni coppia di studenti (composta da un ragazzo ed una ragazza che intendono danzare insieme) ha dovuto registrarsi (altrimenti non avrebbero potuto danzare insieme). I regolamenti della scuola impongono che ogni coppia non possa danzare insieme più di 3 volte. In più, ogni studente non può danzare più di 10 volte in totale. Potete assumere che il ballo duri abbastanza a lungo da permettere a tutti di completare le proprie danze, se le registrazioni lo permettono. Descrivere un algoritmo che, dato in input l'insieme dei maschi e delle femmine e l'insieme delle registrazioni, massimizzi il numero di danze in totale.

Discutere la complessità dell'algoritmo proposto.

Siete responsabile dei corsi di nuoto organizzati da una piscina pubblica. I corsi vengono gestiti settimanalmente.

Ogni corso dura un'ora. Ogni corso può avere al massimo 6 bambini.

Ogni corso ha un orario di inizio che può andare dalle 9 alle 18 (per un totale di dieci corsi al giorno).

Ogni bambino sceglie a quanti corsi partecipare alla settimana, ma questo numero non può essere superiore a 5 e non si può fare lezione più di due volte lo stesso giorno. Poiché i poveri bambini moderni sono stressati da mille altri corsi (calcio, musica, etc), ogni bambino può dare un certo numero di preferenze (ore in cui può partecipare ad un corso). Ogni corso deve essere insegnato da un insegnante. Ogni insegnante può dare un certo numero di disponibilità (ore in cui può insegnare un corso).

Infine, ogni bambino che segue un corso paga 10 euro. Descrivere un algoritmo che di assegnamento bambini-corsi-insegnanti che massimizzi il guadagno, descrivendo per bene input, output. Discutere la complessità.

Quadrato binario

Sia A una matrice $n \times n$ di valori booleani 0/1. Scrivere un algoritmo che prenda in input la matrice A e la sua dimensione n , e restituisca la dimensione del più grande quadrato composto da valori 1 contenuto nella matrice. Ad esempio, nella matrice seguente, i quadrati di dimensione massima sono grandi 4×4 (ve ne sono due, di cui uno evidenziato in rosso).

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

Spoiler alert!

Sequenza specchiata

Non sono necessarie particolari tecniche per risolvere questo esercizio; è sufficiente fare una scansione di costo lineare $\Theta(n)$.

```
int mirror(int [] V, int n)


---


int count0, count1 = 0
int largest = 0
for i = 1 to n do
    if V[i] == 0 then
        if count1 > 0 then
            count0 = 0
            count1 = 0
        count0 = count0 + 1
    else
        count1 = count1 + 1
        largest = max(largest, 2 · min(count0, count1))
return largest
```

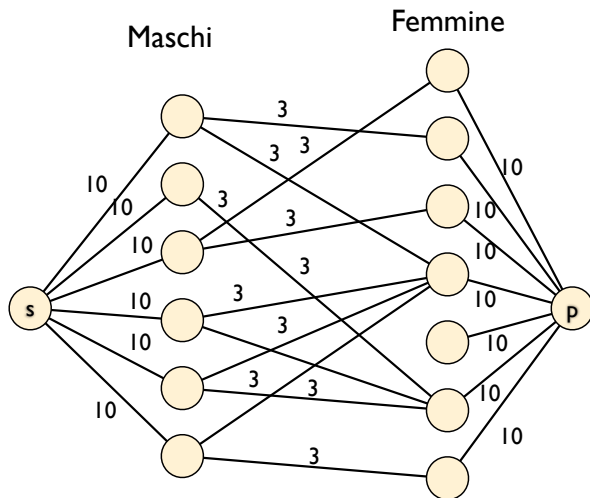
Aggiornamento del flusso massimo

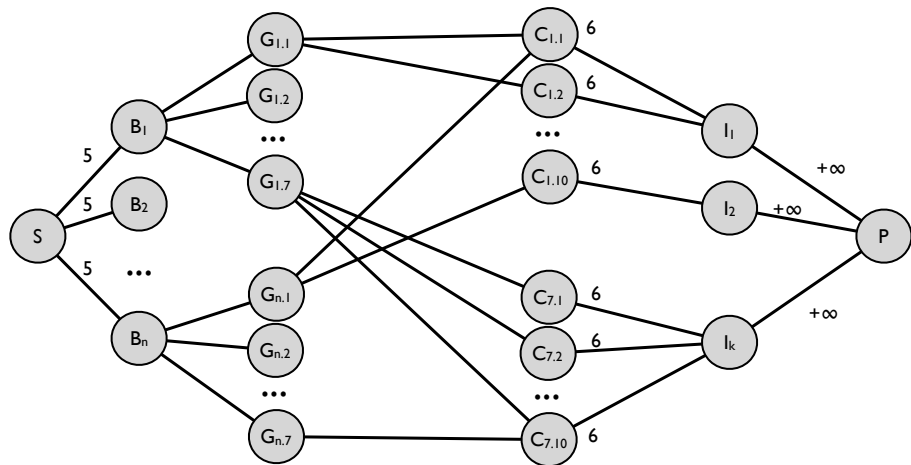
```
maxFlow(int[][] c, int n, int s, int p, int[][] f, int u, int v)
```

```
int[][] r = int[1...n][1...n]           % Rete residua
int[][] g = int[1...n][1...n]           % Cammino aumentante
c[u][v] = c[u][v] + 1
r = c - f
g = cammino-aumentante(r, n, s, p)
f = f + g
```

Nota: Operazioni come $f = f + g$ si intendono come somma, elemento per elemento, delle matrici e hanno costo $O(n^2)$.

Ballo di fine anno





Quadrato binario (Compito 10/09/12)

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

1	0	1	0
1	1	1	1
0	1	2	2
1	1	2	3

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)

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

Quadrato binario

```
int maxSquare(boolean[][] A, int n)
```

```
int[][] DP = new int[1...n][1...n]  
for i = 1 to n do  
    DP[i][1] = A[i][1]  
    DP[1][i] = A[1][i]  
  
    for i = 2 to n do  
        for j = 2 to n do  
            if A[i][j] == 0 then  
                DP[i][j] = 0  
            else  
                DP[i][j] = min(DP[i-1][j], DP[i-1][j-1], DP[i][j-1]) + 1  
  
return max(DP, n)           % Return max value in matrix,  $\Theta(n^2)$ 
```
