Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Trubian, S. Aguzzoli

Appello del 10 Febbraio 2004

Progetto "Filtri"
Consegna entro il 24 Febbraio 2004

Il problema

Obiettivo del progetto è studiare la distorsione di un segnale passante per filtri rettangolari disposti su un piano. Chiamiamo *piano* l'insieme dei punti

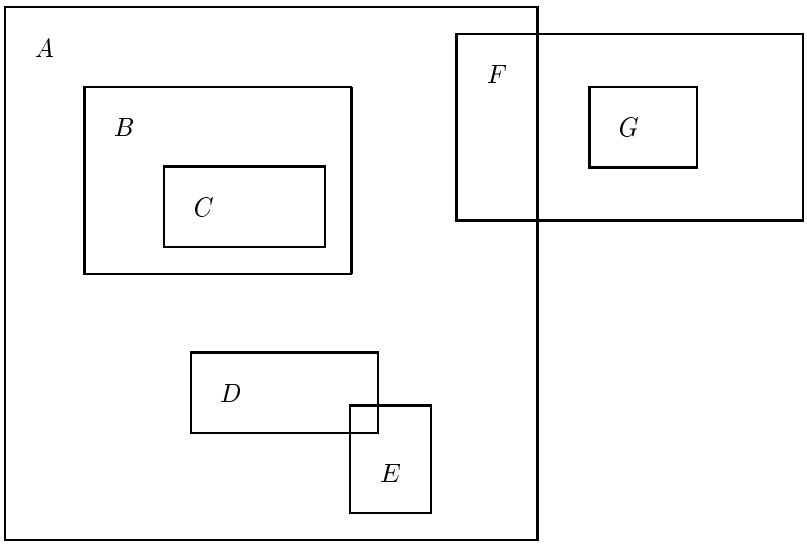
$$\{ (x, y) \in \mathbb{N} \times \mathbb{N} \quad | \quad x \geq 0, \quad y \geq 0\}.$$

Usiamo la notazione $R(a, b, c, d)$ per denotare il rettangolo il cui vertice in basso a sinistra ha coordinate (a, b) e il vertice in alto a destra ha coordinate (c, d), dove $0 \leq a < c$ e $0 \leq b < d$. Formalmente:

$$R(a, b, c, d) = \{(x, y) \in \mathbb{N}^2 \mid a \leq x \leq c, \ b \leq y \leq d\}.$$

Un rettangolo R è *incluso propriamente* in un rettangolo S se $R \subsetneq S$ (visti come insiemi di punti).

Una *catena da R_1 a R_h* è una sequenza di rettangoli $R_1, R_2, \ldots, R_h$ tali che R_i è incluso propriamente in R_{i+1} per $1 \leq i < h$; chiamiamo h la *lunghezza* di tale catena. Il *grado di annidamento* di un rettangolo R è la lunghezza della più lunga catena costituita da rettangoli inclusi propriamente in R. Il grado di annidamento del piano è la lunghezza della più lunga catena costituita da rettangoli contenuti nel piano. Si supponga ad esempio che nel piano siano presenti i rettangoli A, B, C, D, E, F, G disposti come nella figura qui sotto.



Il rettangolo A ha grado di annidamento 2 (la più lunga catena è C, B); B e F hanno grado di annidamento 1; C, D, E e G hanno grado di annidamento 0. Il piano ha grado di annidamento 3 (la più lunga catena è C, B, A).

Supponiamo che ogni punto del piano sia attraversato da uno stesso *segnale* rappresentato da una stringa binaria di lunghezza arbitraria. Un segnale σ è rappresentato come $b_1 b_2 \cdots b_n$ ($n \geq 0$), dove $b_1 = 1$ e $b_i \in \{0, 1\}$ per ogni $i = 2, \ldots, n$. Nel caso $n = 0$ si ottiene la stringa nulla che rappresenta l'assenza di segnale.

I rettangoli posti sul piano costituiscono dei *filtri* che modificano il segnale passante per i punti del piano coperti dai filtri medesimi. La sovrapposizione (anche parziale) di due o più filtri ha un effetto cumulativo sulla *distorsione* del segnale passante per i punti coperti. Ogni filtro è descritto, oltre che dalla sua *forma* rettangolare, da una stringa binaria, eventualmente nulla, che specifica come viene distorto il segnale. Ovviamente una distorsione nulla non ha alcun effetto sul segnale.

Sia quindi (x, y) un punto nel piano e sia $\sigma = b_1 b_2 \cdots b_n$ il segnale che attraversa il piano. Sia $\{R_1, \ldots, R_k\}$ l'insieme dei filtri che contengono il punto (x, y) e sia, per ogni $i = 1, \ldots, k$, τ_i la distorsione associata al filtro R_i. Il segnale $\rho(x, y)$ fuoriuscente dal punto (x, y) è definito come la differenza (binaria) tra σ e la somma (binaria) dei τ_i, troncata a 0:

$$\rho(x, y) = \max(0, \sigma - (\tau_1 + \cdots + \tau_k)).$$

Il caso $\rho(x, y) = 0$ corrisponde all'assenza di segnale (i filtri oscurano completamente il segnale). Si tenga presente che le stringhe binarie possono essere molto lunghe, quindi i loro valori numerici possono non essere rappresentabili con i tipi aritmetici del C.

Per esempio, supponiamo che i rettangoli che contengono (x, y) siano R_1, R_2, R_3 e che abbiano distorsione $\tau_1 = 1101001$, $\tau_2 = 111$, $\tau_3 = 100001000$ e che il piano sia attraversato dal segnale $\sigma = 111111101101$. Allora il segnale fuoriuscente da (x, y) sarà 111001110101. Se invece il segnale del piano è descritto dalla stringa $\sigma = 11111$, allora il segnale fuoriuscente da (x, y) è nullo ed è rappresentato dalla stringa nulla.

Si richiede di implementare una struttura dati efficiente che permette di eseguire le operazioni seguenti (si tenga presente che la minima porzione rettangolare di piano contenente tutti i filtri può essere molto grande rispetto al numero di filtri presenti, quindi *non è sicuramente efficiente rappresentare il piano mediante una matrice*).

Inizialmente il piano non contiene alcun filtro e il segnale che lo attraversa è nullo.

- **segnale**(σ)

  Pone il segnale che attraversa il piano uguale alla stringa binaria σ.

- **filtro**(a, b, c, d, σ)

  Introduce il filtro di forma $R(a, b, c, d)$ con distorsione σ.

- **elimina**(a, b, c, d)

  Elimina dal piano tutti i filtri di forma $R(a, b, c, d)$.

- **grado**(a, b, c, d)

  Stampa il grado di annidamento di $R(a, b, c, d)$ (si noti che questo valore non dipende dal numero di filtri di forma $R(a, b, c, d)$).

- **gradopiano**()

  Stampa il grado di annidamento del piano.

- **punto**(x, y)

  Stampa il carattere $>$ seguito sulla stessa linea dalla stringa descrivente il segnale fuoriuscente dal punto (x, y) (se tale segnale è nullo, viene stampato solo il carattere $>$).

Specifiche di implementazione

Il programma deve leggere dallo standard input (`stdin`) una sequenza di linee (separate da `\n`), ciascuna delle quali corrisponde a una linea della prima colonna della Tabella 1, dove a, b, c, d sono numeri naturali e σ è una stringa binaria, con la cifra più significativa uguale a 1. Si noti che la stringa binaria rappresentante un segnale nullo è la stringa vuota. I vari elementi sulla linea sono separati da uno o più spazi. Quando una linea è letta viene eseguita l'operazione ad essa associata e viene stampato l'eventuale output prodotto dall'esecuzione dell'operazione associata; tutte le operazioni di stampa sono effettuate sullo standard output (`stdout`) e ogni operazione deve iniziare su una nuova linea.

Linea di input	Operazione
`s` σ	**segnale** (σ)
`f` a b c d σ	**filtro**(a, b, c, d, σ)
`e` a b c d	**elimina**(a, b, c, d)
`g` a b c d	**grado**(a, b, c, d)
`G`	**gradopiano**()
`p` a b	**punto**(a, b)
`q`	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

Note

1. Non devono essere presenti vincoli sulla dimensione del piano, dei rettangoli e della lunghezze dei segnali. Non si richiede – anzi si sconsiglia – l'uso di grafica, se non per test personali: in modo particolare, non si usi `conio.h` e neppure `clrscr()`.
2. Per semplicità si suppone che l'input sia sempre conforme alle specifiche di Tabella 1, per cui non è necessario controllare la correttezza dell'input.

Esempio

Si supponga che le linee di input siano:

```
f 2 2 11 10 1101
f 4 4 9 9 11001
s
p 20 20
f 4 4 9 9
f 4 6 8 8 11
f 8 3 10 5 11101001
f 10 7 14 9 111000101
f 10 7 12 12 111000101
```

```
g 2 2 11 10
g 4 4 9 9
g 10 7 14 9
G
p 8 4
s 111001111
p 8 4
p 6 6
p 11 8
e 4 4 9 9
f 10 7 12 9 1000
s 111111111111
g 2 2 11 10
g 10 7 14 9
G
p 11 8
q
```

L'output prodotto dal programma deve essere:

```
>
2
1
0
3
>
> 11000000
> 110100110
>
1
1
2
> 110001100000
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro il 24 Febbraio 2004. La discussione del progetto e l'esame orale si svolgeranno il 1 Marzo 2004 alle 14 in auletta 4.

Occorre presentare:

1. il codice sorgente (rigorosamente ANSI C, compilabile con **gcc**);
2. una sintetica relazione (formato pdf o rtf) che illustra le strutture dati utilizzate e analizza il costo delle diverse operazioni richieste dalla specifica.

I due o più file (file sorgenti C + relazione) devono essere contenuti in un unico file `.zip` il cui nome dovrà essere `cognome.zip`. La relazione e il codice devono riportare il vostro nome, cognome e matricola.

Una copia cartacea della relazione e del codice deve inoltre essere consegnata al dr. Aguzzoli sempre entro il 24 Febbraio 2004 (lasciandola eventualmente nella sua casella postale presso il dipartimento in via Comelico).

Si ricorda infine di presentarsi alla prova orale con una copia stampata della relazione e del codice.

Per ogni ulteriore chiarimento:
E-mail: `aguzzoli@dsi.unimi.it`
Ricevimento: il mercoledì, ore 15-16, stanza S204.

Avvisi

La versione aggiornata del progetto è pubblicata in .pdf sul sito:
`http://homes.dsi.unimi.it/~aguzzoli/algo.htm`.
Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto.

Lo svolgimento del progetto è una prova d'esame da svolgere *individualmente*.

Si richiede allo studente di effettuare un adeguato collaudo del proprio progetto su numerosi esempi diversi per verificarne la correttezza e valutarne le prestazioni.