

Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Trubian, S. Aguzzoli

Appello del 14 Settembre 2004

Progetto "Zoom"
Consegna entro il 3 Ottobre 2004

Il problema

Obiettivo del progetto è studiare le configurazioni di elementi quadrati disposti sul piano, detti celle, ognuna delle quali può essere libera oppure occupata. L'osservazione del piano avviene a risoluzioni diverse: quando si osserva il piano con risoluzione k , si riescono a distinguere solo celle il cui lato misura k unità (per cui 1 è la risoluzione massima). Immaginiamo pertanto che, quando lo osserviamo a risoluzione k , il piano sia suddiviso da una griglia di celle quadrate di lato k . Una cella di lato k è ritenuta occupata se contiene un numero sufficiente di celle occupate di lato 1.

Formalmente, chiamiamo *piano* l'insieme dei punti

$$\{ (x, y) \in \mathbb{N} \times \mathbb{N} \},$$

dove \mathbb{N} denota l'insieme dei numeri naturali $\{0, 1, 2, \dots\}$.

Dato $0 < k \in \mathbb{N}$ e una coppia di naturali $(a, b) \in \mathbb{N} \times \mathbb{N}$, la k -cella $\text{Cella}(k; a, b)$ è il quadrato di lato k il cui vertice in basso a sinistra ha coordinate (ka, kb) :

$$\text{Cella}(k; a, b) = \{ (x, y) \in \mathbb{N} \times \mathbb{N} \mid ka \leq x \leq k(a+1), kb \leq y \leq k(b+1) \}.$$

Ad esempio,

$$\text{Cella}(1; 3, 5) = \{(3, 5), (3, 6), (4, 5), (4, 6)\},$$

mentre

$$\text{Cella}(2; 3, 5) = \{(6, 10), (6, 11), (6, 12), (7, 10), (7, 11), (7, 12), (8, 10), (8, 11), (8, 12)\}.$$

Ogni k -cella può essere *occupata* oppure *libera*. Una 1-cella è occupata su impostazione diretta dell'utente.

Fissato il *parametro di riempimento*, che è un numero razionale p/q , per $0 < p \leq q \in \mathbb{N}$, una k -cella è occupata se e solo se contiene almeno

$$\frac{p}{q}k^2$$

1-celle occupate.

Il *peso* di una k -cella è dato dal numero di 1-celle occupate che essa contiene. Si noti che il peso di una k -cella libera è minore di pk^2/q .

Nell'esempio di Figura 1, se il parametro di riempimento vale $1/3$, allora le 3-celle occupate sono:

$$\text{Cella}(3; 0, 1), \text{Cella}(3; 1, 1), \text{Cella}(3; 1, 2), \text{Cella}(3; 3, 1),$$

il cui peso rispettivamente è 4, 3, 3, 4.

Diciamo che due k -celle sono *adiacenti* se hanno in comune almeno un punto (quindi, ogni k -cella è adiacente a se stessa).

In Figura 1, le 3-celle $\text{Cella}(3;0,1)$ e $\text{Cella}(3;1,1)$ sono adiacenti, $\text{Cella}(3;1,1)$ e $\text{Cella}(3;1,2)$ sono adiacenti, e anche $\text{Cella}(3;0,1)$ e $\text{Cella}(3;1,2)$ sono adiacenti. Al contrario $\text{Cella}(3;3,1)$ non è adiacente ad alcuna delle tre 3-celle prima menzionate.

Un k -cammino da A_1 a A_h è una sequenza di k -celle occupate A_1, \dots, A_h tali che A_i è adiacente a A_{i+1} per $1 \leq i \leq h-1$. Diciamo che un insieme \mathcal{P} di k -celle occupate è *connesso* se, per ogni $A, B \in \mathcal{P}$, esiste un k -cammino da A a B contenuto in \mathcal{P} . Un k -blocco è un insieme di k -celle occupate \mathcal{A} tale che \mathcal{A} è connesso e, per ogni k -cella occupata $P \notin \mathcal{A}$, $\mathcal{A} \cup \{P\}$ non è connesso (quindi un k -blocco è un insieme connesso massimale di k -celle occupate).

L'area di un k -blocco è dato dal numero di k -celle appartenenti al blocco.

Il peso di un k -blocco è dato dalla somma dei pesi delle k -celle appartenenti al blocco.

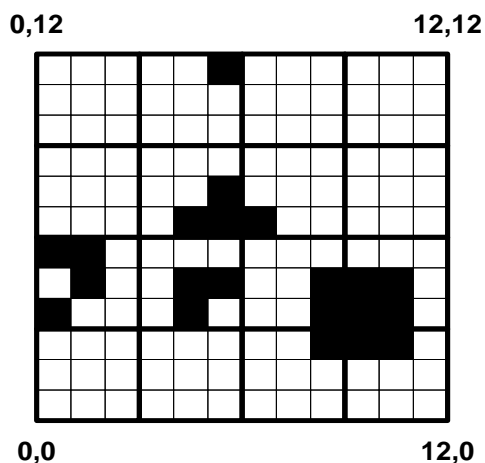


Figura 1: Esempio

Nell'esempio di Figura 1, assumendo che il parametro di riempimento valga $1/3$, vi sono 2 3-blocchi:

$$\{\text{Cella}(3;0,1), \text{Cella}(3;1,1), \text{Cella}(3;1,2)\}, \quad \{\text{Cella}(3;3,1)\},$$

i cui pesi rispettivamente valgono 10 e 4. Impostando il fattore di riempimento al valore $2/9$ abbiamo un unico 3-blocco così formato:

$$\{\text{Cella}(3;0,1), \text{Cella}(3;1,1), \text{Cella}(3;1,2), \text{Cella}(3;2,1), \text{Cella}(3;3,0), \text{Cella}(3;3,1)\},$$

di peso 18. Si noti che non vi è alcun 3-blocco quando il parametro di riempimento vale $1/2$. A risoluzione massima ($k=1$), qualunque sia il valore del parametro di riempimento, vi sono 5 1-blocchi.

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

- **crea** (p, q, k)

Crea un piano vuoto (eliminando l'eventuale piano già esistente), imposta il parametro di riempimento al valore $\min\{p/q, q/p\}$ e la risoluzione corrente a k .

- **riempimento** (p, q)
Imposta il parametro di riempimento al valore $\min\{p/q, q/p\}$.
- **risoluzione** (k)
Imposta la risoluzione corrente a k .
- **inserisci** (x, y)
Occupi la 1-cella $\text{Cella}(1; x, y)$. In caso tale cella sia già occupata non viene eseguita alcuna operazione.
- **elimina** (x, y)
Detta k la risoluzione corrente, se la k -cella $\text{Cella}(k; x, y)$ risulta occupata, tutte le 1-celle contenute in $\text{Cella}(k; x, y)$ vengono liberate. Se invece $\text{Cella}(k; x, y)$ risulta libera non viene eseguita alcuna operazione (si noti che una k -cella libera può contenere alcune 1-celle occupate: basta che il loro numero non superi pk^2/q).
- **area** (x, y)
Calcola l'area del k -blocco contenente $\text{Cella}(k; x, y)$, dove k è la risoluzione corrente. Si noti che se $\text{Cella}(k; x, y)$ è libera, essa non appartiene ad alcun blocco, e l'area richiesta è 0.
- **peso** (x, y)
Calcola il peso del k -blocco contenente $\text{Cella}(k; x, y)$, dove k è la risoluzione corrente. Si noti che se $\text{Cella}(k; x, y)$ è libera, essa non appartiene ad alcun blocco, e il peso richiesto è 0.
- **blocchi** ()
Calcola il numero di k -blocchi presenti nel piano, dove k è la risoluzione corrente.

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 x, y sono numeri naturali ≥ 0 , mentre p, q, k sono naturali maggiori di 0. I vari elementi sulla linea sono separati da uno o più spazi. Quando una linea è letta, viene eseguita l'operazione associata; le operazioni di stampa sono effettuate sullo standard output (`stdout`), e ogni operazione deve iniziare su una nuova linea.

Note

1. Non devono essere presenti vincoli sulla dimensione del piano e sul numero di celle (se non quelli determinati dal tipo di dato intero). 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. Per leggere l'input si usino le funzioni standard ANSI C `getchar()` e/o `scanf()`.

LINEA DI INPUT	OPERAZIONE
<i>c p q k</i>	crea (p, q, k)
<i>r p q</i>	riempimento (p, q)
<i>R k</i>	risoluzione (k)
<i>i x y</i>	inserisci (x, y)
<i>e x y</i>	elimina (x, y)
<i>a x y</i>	area (x, y)
<i>p x y</i>	peso (x, y)
<i>b</i>	blocchi ()
<i>f</i>	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

Esempio

Si supponga che le linee di input siano:

```

c 25 3 5
i 2 2
i 3 1
i 3 3
i 4 1
i 5 1
i 8 4
i 9 3
i 9 4
i 10 2
i 11 2
i 13 8
i 14 4
i 14 5
i 14 7
i 15 4
i 15 5
i 15 7
i 19 2
i 19 3
i 19 4
i 20 2
i 21 1
i 22 0

```

```
i 25 1
a 1 0
p 2 1
b
r 25 4
a 1 0
p 2 1
a 3 0
p 3 0
b
e 0 0
e 1 0
e 2 0
i 9 0
i 12 3
i 23 1
i 24 0
R 3
b
a 4 2
p 4 2
r 13 39
b
a 3 0
p 4 1
a 7 0
p 7 0
a 5 1
R 1
b
a 23 1
p 19 3
a 9 1
p 15 7
a 9 0
f
```

L'output prodotto dal programma deve essere:

```
6
21
1
2
0
1
4
2
1
8
19
```

2
2
6
1
3
0
5
9
9
0
3
1

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo aguzzoli@dsi.unimi.it entro il 3 Ottobre 2004 (incluso). La discussione del progetto e l'esame orale si svolgeranno in data e luogo da specificarsi (consultare al riguardo il sito: <http://homes.dsi.unimi.it/~trubian/studenti.htm>).

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 3 Ottobre 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.

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

Lo svolgimento del progetto è una prova d'esame da svolgere **individualmente**. I progetti giudicati frutto di **collaborazioni** saranno **estromessi** d'ufficio dalla valutazione.