Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Goldwurm, S. Aguzzoli

Appello dell'1 luglio 2003

Progetto "Celle Colorate"
Consegna entro il 17 luglio 2003

Il problema

Obiettivo del progetto è studiare le configurazioni di insiemi di celle colorate su un piano, e la loro influenza sulle celle circonvicine. Il piano è suddiviso in quadrati di lato unitario che chiamiamo *celle*. In un dato istante t, ogni cella può essere *colorata* o *vuota*.

Formalmente, chiamiamo *piano* l'insieme dei punti

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

Una *cella* è un quadrato il cui lato ha dimensione unitaria; più precisamente, data una coppia di naturali (a,b), una cella in *posizione* (a,b) è data dall'insieme dei punti (vertici del quadrato):

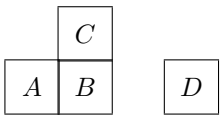
$$\text{Cella}(a,b) \ = \ \{\ (x,y) \quad | \quad a \leq x \leq a+1,\ b \leq y \leq b+1\}.$$

Ad esempio, $\text{Cella}(3,5) \ = \ \{(3,5),(3,6),(4,6),(4,5)\}$

Ogni cella può essere *colorata* su richiesta dell'utente. L'insieme dei colori disponibili è l'insieme delle stringhe sull'alfabeto $\{a,b,\ldots,z\}$.

La *colorazione* di una cella avviene specificando una terna (x,y,α), dove $x,y \in \mathbb{N}$ rappresentano le coordinate della cella da colorare, mentre α è la stringa che rappresenta il colore.

Diciamo che due celle sono *adiacenti* se hanno in comune almeno un punto (quindi, ogni cella è adiacente a se stessa); nella figura qui sotto le celle A e B sono adiacenti, B e C sono adiacenti, e anche A e C sono adiacenti. Al contrario D non è adiacente ad alcuna delle altre tre celle.



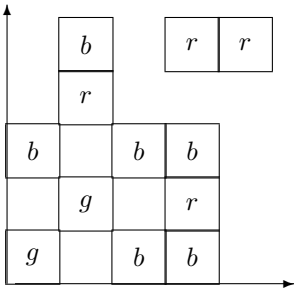
Un *cammino* da A_1 a A_h è una sequenza di celle colorate $A_1,\ldots,A_h$ tali che A_i è adiacente a A_{i+1} per $1 \leq i \leq h-1$. Diciamo che un insieme $\mathcal{P}$ di celle colorate è *connesso* se, per ogni $A,B \in \mathcal{P}$, esiste un cammino da A a B contenuto in $\mathcal{P}$. Un *blocco*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

Nell'esempio sopra $\{A\}$, $\{B\}$, $\{C\}$, $\{A,B\}$, $\{B,C\}$, $\{A,B,C\}$, $\{A,C\}$ sono connessi, mentre $\{A,D\}$, $\{B,D\}$, $\{A,B,C,D\}$, $\{B,C,D\}$, etc., non sono connessi. I blocchi sono $\{A,B,C\}$ e $\{D\}$.

Il *blocco di appartenza* di $\text{Cella}(x,y)$ è il blocco che contiene Cella (x,y).

Un *blocco omogeneo* è un insieme di celle $\mathcal{A}$ dello stesso colore α tale che $\mathcal{A}$ è connesso e, per ogni altra cella $P \notin \mathcal{A}$ di colore α, $\mathcal{A} \cup \{P\}$ non è connesso.

Il *blocco omogeneo di appartenenza* di Cella(x, y) è il blocco omogeneo che contiene Cella(x, y).
Si consideri l'esempio seguente, dove i colori considerati sono r, g, b



Si noti che vi sono 2 blocchi: il blocco di appartenenza della cella colorata posizionata in Cella$(0, 0)$ contiene 10 celle, mentre il blocco di appartenenza della cella in Cella$(3, 4)$ conta 2 celle Il blocco omogeneo di appartenza della cella posizionata in Cella$(0, 0)$ contiene 2 celle, mentre il blocco omogeneo di appartenenza della cella in Cella$(2, 0)$ ha 2 celle.

L'*intorno* della cella s è l'insieme di tutte le celle adiacenti ad s e differenti da s.

È possibile definire *regole di ricolorazione* del tipo

$$k_1\alpha_1 + k_2\alpha_2 + \cdots + k_n\alpha_n \to \beta$$

dove $\alpha_1, \ldots, \alpha_n$ sono stringhe sull'alfabeto $\{a, b, \ldots, z\}$ tutte differenti tra loro, β è una stringa sull'alfabeto $\{a, b, \ldots, z\}$ e $k_i \in \{1, \ldots, 8\}$ sono tali che $\sum_{i=1}^{n} k_i \leq 8$.

La regola è *applicabile* a un a cella s se l'intorno di s contiene *almeno* k_i celle di colore α_i per ogni $i \in \{1, \ldots, n\}$. L'effettiva *applicazione* della regola determina la (ri)colorazione della cella s con il colore β.

Si noti che l'applicabilità di una regola a una cella s non dipende dal colore di s, che può essere anche vuota.

Le regole di ricolorazione devono venire specificate esplicitamente attraverso un opportuno comando.

Le regole sono inserite dall'utente e vengono memorizzate in un elenco in ordine cronologico di inserimento (vale a dire, la regola inserita più recentemente è l'ultima dell'elenco).

L'elenco ordinato delle regole descrive come le celle vengono (ri)colorate. Qualora venga richiesto di effettuare una ricolorazione di una cella, fra tutte le regole applicabili alla cella stessa viene selezionata ed applicata solo la prima di queste a comparire nell'elenco.

È possibile *ricolorare un blocco* in questo modo: per ogni cella colorata s appartenente al blocco si applica la prima regola dell'elenco applicabile a s rispetto all'intorno di s *cosí come si presenta all'inizio della procedura di ricolorazione del blocco*. Dunque, eventuali ricolorazioni di celle del blocco appartenenti all'intorno non influiscono sulla ricolorazione di s.

Riprendendo l'esempio precedente supponiamo che le regole inserite siano nell'ordine:

$$\begin{aligned} &2g + 1b \to z \\ &1g + 2b \to w \\ &1b + 1r \to y \\ &2b + 1r \to g \\ &1b + 1g + 1r \to t. \end{aligned}$$

Ricoloriamo Cella$(1, 1)$: il suo intorno contiene 1 cella di colore g e 3 di colore b. Quindi la prima regola applicabile è la seconda dell'elenco. La sua applicazione colora Cella$(1, 1)$ col colore w. Supponiamo ora

di voler colorare Cella$(3,3)$, che attualmente è vuota. L'intorno contiene 2 celle di colore b e 2 celle di colore r, dunque la prima regola applicabile è la terza dell'elenco che conferisce a Cella$(3,3)$ il colore y. Si noti che in questo caso anche la quarta regola è applicabile, ma non viene applicata perché è già applicabile la terza.

Riprendendo ora la colorazione iniziale dell'esempio ricoloriamo il blocco di appartenenza di Cella$(1,1)$. Le celle che sono soggette a ricolorazione sono: Cella$(1,1)$ che come già visto assumerà il colore w per applicazione della seconda regola, Cella$(2,0)$, Cella$(2,2)$, Cella$(3,0)$ e Cella$(3,2)$ che assumeranno il colore y per applicazione della terza regola.

Durante l'esecuzione, ogni regola viene applicata un certo di numero di volte: definiamo *uso* di una regola la quantità determinata dal numero totale di celle a cui la regola è stata applicata dal momento della sua definizione. In dettaglio, siano $p \geq 0$ le operazioni di ricolorazione di celle o di ricolorazione di blocco eseguite fino ad ora, e si denoti con $u_i(r)$ per ogni $i \in \{1, \ldots, p\}$ il numero totale di celle a cui si è applicata la regola r durante la i-esima operazione di ricolorazione. Allora il valore attuale dell'uso della regola r è dato da $uso(r) = \sum_{i=1}^{p} u_i(r)$.

Nell'esempio precedente, la ricolorazione del blocco di appartenenza di Cella$(1,1)$ causa l'incremento di 4 unità dell'uso della terza regola e di 1 della seconda regola.

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 tutte le celle colorate può essere molto grande rispetto al numero di celle colorate, quindi *non è sicuramente efficiente rappresentare il piano mediante una matrice*).

- **colora**(x, y, α)

  Colora Cella(x, y) di colore α, qualunque sia lo stato di Cella(x, y) prima dell'operazione.

- **elimina**(x, y)

  Elimina dal piano la cella eventualmente contenuta in Cella(x, y). Vale a dire che dopo l'operazione Cella(x, y) sarà vuota.

- **area**(x, y)

  Calcola il numero totale di celle contenute nel blocco di appartenenza della cella in Cella(x, y). Se Cella(x, y) è vuota, restituisce 0.

- **area_omo**(x, y)

  Calcola il numero totale di celle contenute nel blocco omogeneo di appartenenza della cella in Cella(x, y). Se Cella(x, y) è vuota, restituisce 0.

- **definisci**$(k_1, \alpha_1, k_2, \alpha_2, \ldots, k_n, \alpha_n, \beta)$

  Definisce la regola di ricolorazione $k_1\alpha_1 + k_2\alpha_2 + \cdots + k_n\alpha_n \to \beta$ e la inserisce in coda all'elenco delle regole.

- **ricolora**(x, y)

  Applica a Cella (x, y) la prima regola applicabile dell'elenco, ricolorando la cella. Se nessuna regola è applicabile, non viene eseguita alcuna operazione.

- **ricolora_blocco**(x, y)

  Applica a ogni cella Cella(a, b) del blocco di appartenenza di Cella(x, y) l'operazione **ricolora**(a, b).

- **stampa**

  Stampa l'elenco ordinato delle regole di ricolorazione.

- **ordina**

  Ordina l'elenco delle regole di ricolorazione in ordine inverso rispetto all'uso delle regole stesse: la regola con uso maggiore diventa l'ultima dell'elenco. Se due regole hanno uso uguale mantengono il loro ordine relativo.

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 $x, y, k_1, \ldots, k_n$ sono numeri naturali e $\alpha_1, \ldots, \alpha_n, \beta$ sono stringhe sull'alfabeto $\{a, b, \ldots, z\}$ e 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
`c` x y α	**inserisci** (x, y, α)
`e` x y	**elimina**(x, y)
`a` x y	**area**(x, y)
`A` x y	**area_omo**(x, y)
`d` $k_1\, \alpha_1\, k_2\, \alpha_2 \cdots k_n\, \alpha_n\, 0\, \beta$	**definisci** $(k_1, \alpha_1, k_2, \alpha_2, \ldots, k_n, \alpha_n, \beta)$
`r` x y	**ricolora**(x, y)
`R` x y	**ricolora_blocco**(x, y)
`s`	**stampa**
`o`	**ordina**
`f`	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

Per quanto riguarda la stampa dell'elenco di regole, occorre stampare una regola per riga nello stesso formato previsto per l'input.

Si noti che 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()`.

Nota: 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:

```
c 1 1 arancio
```

```
c 2 2 blu
c 3 1 arancio
c 5 2 arancio
c 5 3 blu
c 5 4 rosso
c 6 3 rosso
d 2 arancio 0 blu
d 1 rosso 1 blu 0 giallo
d 1 arancio 1 giallo 0 arancio
r 2 1
r 5 4
a 1 1
A 1 1
a 2 1
A 2 1
c 1 2 giallo
d 2 blu 1 giallo 0 rosso
d 1 blu 0 rosso
A 2 2
A 3 1
A 1 1
R 2 2
A 2 2
A 3 1
A 1 1
e 5 2
o
s
A 5 4
r 6 4
A 5 4
f
```

L'output prodotto dal programma deve essere:

```
4
1
4
2
2
1
1
2
1
2
d 1 arancio 1 giallo 0 arancio
d 1 rosso 1 blu 0 giallo
d 2 blu 1 giallo 0 rosso
d 1 blu 0 rosso
```

```
d 2 arancio 0 blu
1
2
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro il 17 luglio 2003. La discussione del progetto e L'esame orale si svolgeranno il 22 luglio 2003 in aula 5 alle 9:00.

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 17 luglio 2003 (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*.