Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Goldwurm, S. Aguzzoli

Appello del 4 giugno 2003

Progetto "Mosaico"
Consegna entro il 22 giugno 2003

Il problema

Obiettivo del progetto è studiare le configurazioni di insiemi di tessere colorate su un piano. Il piano è suddiviso in quadrati di lato unitario che chiamiamo *celle*.

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 occupata su richiesta dell'utente da una *tessera colorata*. L'insieme dei colori disponibili è l'insieme delle stringhe sull'alfabeto $\{a, b, \ldots, z\}$.

L'*inserimento* di una tessera colorata avviene specificando una terna (x, y, α), dove $x, y \in \mathbb{N}$ rappresentano le coordinate della cella dove posizionare la tessera, mentre α è la stringa che rappresenta il colore della tessera stessa.

È possibile definire *regole di ricolorazione* del tipo

$$\alpha + \beta \to \gamma$$

dove α, β, γ sono stringhe sull'alfabeto $\{a, b, \ldots, z\}$.

Per mezzo della regola $\alpha + \beta \to \gamma$ è possibile ricolorare una tessera di colore α sovrapponendovi (cioè inserendo nella stessa posizione) una tessera di colore β, ottenendo una tessera risultante di colore γ. Si noti che una sovrapposizione, o meglio ricolorazione, è possibile solo se la regola opportuna è già stata esplicitamente definita dall'utente tramite un opportuno comando (vedi sotto).

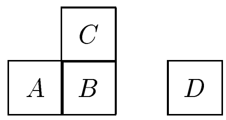
Assumiamo la commutatività delle regole di ricolorazione, per cui la definizione della regola $\alpha + \beta \to \gamma$ implica la definizione implicita della regola $\beta + \alpha \to \gamma$. Ci riferiremo ad entrambe come regole di ricolorazione *per* α, β.

Si noti che una cella o non è occupata da alcuna tessera o è occupata da esattamente una tessera, poiché la sovrapposizione è da intendersi come semplice ricolorazione della tessera già presente.

Per semplicità una volta che sia stata definita la regola $\alpha + \beta \to \gamma$ o la regola equivalente $\beta + \alpha \to \gamma$ non sarà più possibile per l'utente modificarla, cioè il sistema dovrà impedire il tentativo di definizione di altre regole della forma $\alpha + \beta \to \delta$ o della forma $\beta + \alpha \to \delta$.

Si noti che la ricolorazione di una tessera in genere ne cambia il colore, ma ciò che si ottiene è sempre una sola tessera del nuovo colore, che potrà a sua volta essere eventualmente ricolorata. Ad esempio: si supponga di aver definito le regole *bianca*+*rossa* → *rosa* e *rosa*+*nera* → *nera* e di aver inserito la tessera $(1, 1, rossa)$. Sovrapponendo a quest'ultima la tessera $(1, 1, bianca)$, otteniamo la tessera $(1, 1, rosa)$. Questa tessera può subire a sua volta una ricolorazione, ad esempio, inserendo la tessera $(1, 1, nera)$ a cui si applica automaticamente la regola di ricolorazione che produce la tessera $(1, 1, nera)$.

Diciamo che due tessere sono *adiacenti* se hanno in comune almeno un punto (quindi, ogni tessera è adiacente a se stessa); nella figura qui sotto le tessere 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 tessere.



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

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 $\{D\}$.

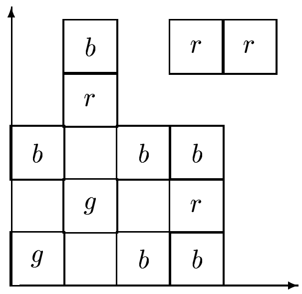
Il *perimetro* di un blocco è dato dall'insieme dei lati sul *bordo* del blocco (ossia, i lati che appartengono a *una sola* cella del blocco); la *lunghezza* del perimetro è data dal numero dei lati che lo compongono. Ad esempio la lunghezza del perimetro del blocco a cui appartiene la cella A nell'esempio precedente è 8.

Il *blocco di appartenza* di una tessera (x, y, α) è il blocco che contiene la cella (x, y).

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

Il *blocco colorato di appartenenza* di una tessera (x, y, α) è il blocco colorato che contiene la cella (x, y).

Si consideri l'esempio seguente, dove i colori considerati sono r, g, b e sia stata definita la regola di ricolorazione $r + g \to b$:



Si noti che vi sono 2 blocchi: il blocco di appartenenza della tessera posizionata in Cella$(0, 0)$ contiene 10 tessere ed ha perimetro di lunghezza 30, mentre il blocco di appartenenza della tessera in Cella$(3, 4)$ conta 2 tessere e ha perimetro 6. Il blocco colorato di appartenza della tessera posizionata in Cella$(0, 0)$ contiene 2 tessere ed ha perimetro 8, mentre il blocco colorato di appartenenza della tessera in Cella$(2, 0)$ ha 2 tessere e perimetro di lunghezza 6. Inserendo la tessera $(1, 1, r)$ si attua una ricolorazione della tessera in Cella$(1, 1)$ che diventa di colore b, mentre il blocco colorato di appartenenza della tessera in Cella$(2, 0)$ ora contiene 6 tessere e la lunghezza del suo perimetro è 20.

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

- **inserisci** (x, y, α)

  Inserisce in Cella(x, y) la tessera di colore α. Se in tale cella è già presente una tessera di colore β, l'operazione viene eseguita se e solo se è già definita una regola di ricolorazione della forma $\alpha + \beta \to \gamma$ o $\beta + \alpha \to \gamma$ per qualche colore γ; in questo caso, l'effetto finale sarà la presenza in Cella(x, y) di una tessera di colore γ. Se invece nessuna regola della forma specificata è definita, viene stampato il messaggio

  `Impossibile combinare` α `con` β`.`

- **elimina**(x, y)

  Elimina dal piano la tessera eventualmente contenuta in Cella(x, y). In caso tale cella non contenga alcuna tessera non viene eseguita alcuna operazione.

- **perimetro**(x, y)

  Calcola la lunghezza del perimetro del blocco di appartenenza della tessera in Cella(x, y).

- **perimetro-col**(x, y)

  Calcola la lunghezza del perimetro del blocco colorato di appartenenza della tessera in Cella(x, y).

- **area**(x, y)

  Calcola il numero totale di tessere contenute nel blocco di appartenenza della tessera in Cella(x, y).

- **area-col**(x, y)

  Calcola il numero totale di tessere contenute nel blocco colorato di appartenenza della tessera in Cella(x, y).

- **definisci**(α, β, γ)

  Definisce la regola di ricolorazione $\alpha + \beta \to \gamma$ (e implicitamente la regola simmetrica $\beta + \alpha \to \gamma$). Se una regola di ricolorazione per α e β è già definita, non viene eseguita alcuna operazione.

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 e α, β, γ 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 associata; le operazioni di stampa sono effettuate sullo standard output (`stdout`), e ogni operazione deve iniziare su una nuova linea.

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:

Linea di input	Operazione
`i` x y α	**inserisci** (x, y, α)
`e` x y	**elimina**(x, y)
`p` x y	**perimetro**(x, y)
`P` x y	**perimetro-col**(x, y)
`a` x y	**area**(x, y)
`A` x y	**area-col**(x, y)
`d` α β γ	**definisci** (α, β, γ)
`f`	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

```
i 1 1 blu
i 2 1 blu
i 2 2 rosso
i 3 3 blu
i 5 4 giallo
i 6 4 rosso
a 1 1
A 1 1
p 1 1
P 1 1
i 2 2 giallo
d rosso giallo arancione
d melone arancione blu
i 2 2 giallo
i 2 2 melone
a 1 1
A 1 1
p 1 1
P 1 1
e 2 2
i 4 4 blu
a 4 4
A 4 4
p 4 4
P 4 4
f
```

L'output prodotto dal programma deve essere:

```
4
2
12
6
Impossibile combinare giallo con rosso.
4
4
12
12
4
2
12
8
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro il 22 Giugno 2003. La discussione del progetto e L'esame orale si svolgeranno il 30 Giugno 2003 in aula α 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 22 Giugno 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*.