

# Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Trubian, S. Aguzzoli

Appello del 1 Giugno 2004

Progetto “Specchi”

Consegna entro il 17 Giugno 2004

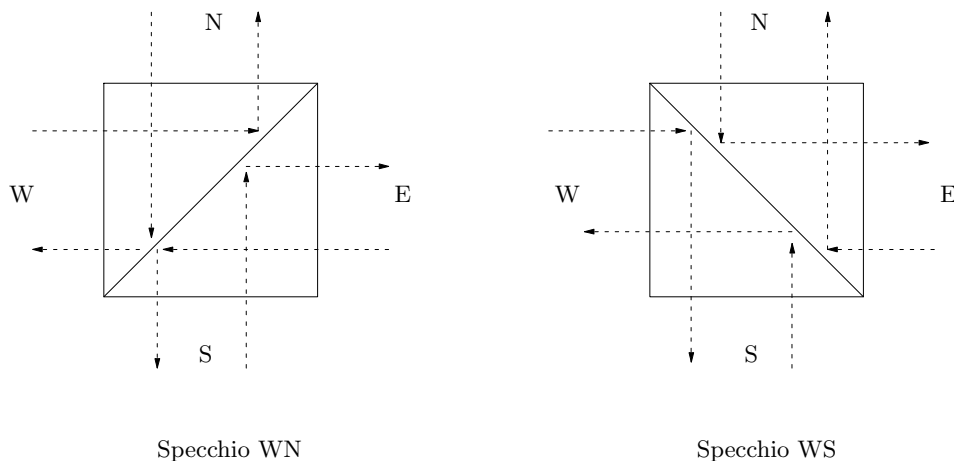
## Il problema

Obiettivo del progetto è studiare il percorso di raggi colorati in un piano su cui sono disposti degli specchi quadrati in grado di modificare direzione e colore dei raggi.

Chiamiamo *piano* l’insieme dei punti

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

Uno *specchio*  $S(x, y, w, z)$  è un quadrato avente come *diagonale* il segmento di estremi  $(x, y)$  e  $(w, z)$ , sulla quale giace il piano riflettente dello specchio. Un *raggio* percorre il piano in direzione parallela all’asse delle ascisse o delle ordinate e la sua traiettoria è modificata di 90 gradi quando incontra uno specchio. Il modo in cui un raggio è deviato dipende dalla posizione della diagonale del quadrato, che può essere di due tipi come rappresentato nella figura qui sotto.



Lo specchio a sinistra (*tipo WN*) devia a *N* i raggi che arrivano da *W*, a *W* i raggi che arrivano da *N*, a *S* i raggi che arrivano da *E* e a *E* quelli che arrivano da *S*, come indicato dalle frecce. Lo specchio a destra (*tipo WS*) si comporta in modo simmetrico. Se il raggio colpisce uno dei quattro vertici dello specchio non viene deviato.

Si suppone che un raggio arrivi da una sorgente posta a distanza infinita dalla regione del piano contenente gli specchi e che percorra il piano in direzione parallela all’asse delle ascisse se proviene da *W* o da *E*, oppure in direzione parallela all’asse delle ordinate se proviene da *S* o da *N*. Quando il raggio incontra uno specchio, il suo cammino viene deviato nel modo sopra specificato. Quando il raggio non incontra più specchi sul suo percorso, il suo cammino si allontana definitivamente dalla regione del piano contenente gli specchi seguendo una traiettoria rettilinea (si noti che non è possibile che si formino dei cicli). Per definire l’*origine* e la *destinazione* di un raggio è sufficiente definire una coppia  $(P, k)$ , dove:

- $P \in \{N, S, E, W\}$  denota il punto cardinale da cui il raggio proviene o verso cui è diretto, mentre  $k \in \mathbb{Z}$ .
- Se  $P \in \{N, S\}$ , il raggio percorre la retta di equazione  $x = k$  parallela all'asse delle ordinate.
- Se  $P \in \{E, W\}$ , il raggio percorre la retta di equazione  $y = k$  parallela all'asse delle ascisse.

Ad esempio, supponiamo che gli specchi nella figura precedente siano rispettivamente  $S(-8, -3, -3, 2)$  e  $S(2, 2, 7, -3)$ . Allora, il raggio di origine  $(N, -6)$  ha destinazione  $(W, -1)$  (infatti incontra la diagonale del primo specchio nel punto  $(-6, -1)$  e il suo cammino prosegue verso  $W$ ); il raggio di origine  $(E, 1)$  ha destinazione  $(N, 3)$ . Il raggio di origine  $(S, -6)$  ha destinazione  $(S, 5)$ ; infatti, tale raggio incontra la diagonale del primo specchio nel punto  $(-6, -1)$  che modifica il suo percorso verso  $E$ , quindi incontra la diagonale del secondo specchio nel punto  $(5, -1)$  che lo devia verso  $S$ , quindi non incontra più specchi. Si noti che il raggio di origine  $(N, 2)$  ha destinazione  $(S, 2)$  in quanto incontra la diagonale del secondo specchio in un vertice, quindi non viene deviato.

Un raggio è caratterizzato anche da un *colore* descritto da una stringa non vuota sull'alfabeto  $\mathcal{A}$  delle lettere minuscole  $\{a, b, c, \dots, z\}$ . Uno specchio  $S$  oltre che modificare la direzione del raggio, ne modifica anche il colore nel modo specificato da *regole* di ricolorazione del tipo:

$$a_1 \rightarrow \sigma_1, a_2 \rightarrow \sigma_2, \dots, a_n \rightarrow \sigma_n$$

dove  $a_1, a_2, \dots, a_n$  sono elementi dell'alfabeto  $\mathcal{A}$  e  $\sigma_1, \sigma_2, \dots, \sigma_n$  sono stringhe finite, eventualmente vuote, su  $\mathcal{A}$ . Supponiamo che un raggio abbia colore  $\alpha = s_1 \dots s_k$  e che la sua direzione venga modificata dallo specchio  $S$ ; allora il nuovo colore  $\beta$  è dato dalla stringa  $\gamma_1 \dots \gamma_k$  dove, per ogni  $i = 1, \dots, k$ , vale:

- se esiste la regola di ricolorazione  $s_i \rightarrow \sigma_i$ , allora  $\gamma_i$  è la stringa  $\sigma_i$  ( $s_i$  è riscritto da  $\sigma_i$ );
- altrimenti,  $\gamma_i$  coincide con  $s_i$  ( $s_i$  rimane inalterato).

Nell'esempio di prima, se al primo specchio sono associate le regole

$$a \rightarrow b, c \rightarrow aba$$

e al secondo le regole

$$a \rightarrow aba, b \rightarrow c, c \rightarrow bba$$

il raggio di origine  $(E, 1)$  e colore  $abc$  ha destinazione  $(N, 3)$  e colore  $abacbbba$ ; il raggio di origine  $(S, -6)$  e colore  $abc$  ha destinazione  $(S, 5)$  e colore  $ccabacaba$ ; il raggio di provenienza  $(N, 2)$  non cambia colore. Diciamo che due specchi (visti come quadrati) *collidono* se la loro intersezione ha area di dimensione non nulla (vale a dire, l'intersezione è un insieme bidimensionale). Assumiamo che due specchi del piano non possano collidere, quindi possono avere in comune solamente un vertice, un lato o una porzione di lato. Quando viene richiesto l'inserimento di uno specchio  $S(x, y, w, z)$  occorre verificare che esso non collida con gli specchi già presenti nel piano. Se questo avviene, occorre modificare la *forma* di  $S$  (ma non il tipo e le regole) nel seguente modo:

- (1) Si determina il minimo rettangolo  $R$  contenente lo specchio  $S(x, y, w, z)$  e tutti gli specchi che collidono con  $S(x, y, w, z)$ .
- (2) Se  $S$  è uno specchio di tipo  $WN$ , si trova il minimo quadrato  $Q$  contenente  $R$  e avente in comune con  $R$  il vertice a  $NE$ . Si inserisce quindi lo specchio di forma  $Q$  e avente le stesse caratteristiche (tipo e regole di ricolorazione) di  $S$ .

- (3) Se  $S$  è uno specchio di tipo  $WS$ , si trova il minimo quadrato  $Q$  contenente  $R$  e avente in comune con  $R$  il vertice a  $SW$ . Si inserisce quindi lo specchio di forma  $Q$  e avente le stesse caratteristiche (tipo e regole di ricolorazione) di  $S$ .

Si noti che l'inserimento di un nuovo specchio nei punti (2) e (3) può richiedere una nuova esecuzione di (1) in quanto il nuovo specchio di forma  $Q$  potrebbe di nuovo collidere con altri specchi; ovviamente, la procedura terminerà in un numero finito di passi con l'inserimento di uno specchio. Quando lo specchio definito in (2) o in (3) viene inserito, vengono eliminati gli eventuali specchi presenti sul piano che collidono con esso. Ad esempio, supponiamo che nel piano siano presenti gli specchi  $A$ ,  $B$  e  $C$  e di voler inserire lo specchio  $S$  di tipo  $WN$  (vedi Figura 1). Poiché  $S$  collide con  $A$  e  $B$ , occorre determinare il

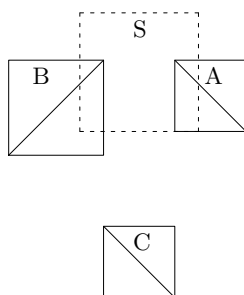


Figura 1:

minimo rettangolo  $R$  che include i quadrati  $A$ ,  $B$  e  $S$  (vedi Figura 2). Essendo  $S$  di tipo  $WN$ , occorre

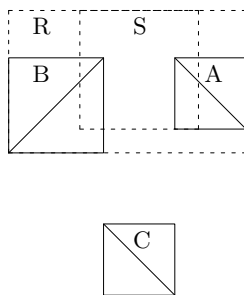


Figura 2:

estendere  $R$  in un quadrato  $Q$  come indicato nel punto 2 (vedi Figura 3). Tuttavia  $Q$  non è ancora inseribile, in quanto  $Q$  collide con  $C$ . Ripetendo la procedura si ottiene lo specchio  $D$  (vedi Figura 4) che contiene gli specchi  $A$ ,  $B$  e  $C$ . Quando  $D$  viene inserito, gli specchi  $A$ ,  $B$  e  $C$  sono eliminati.

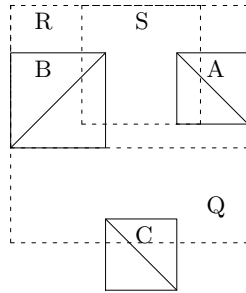


Figura 3:

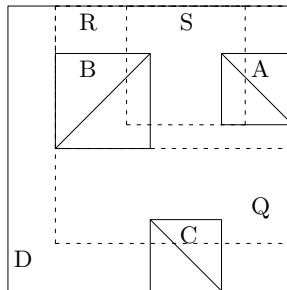


Figura 4:

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

- **raggio** $(P, k, \alpha)$

Stampa la destinazione  $(Q, h)$  e il colore finale  $\beta$  del raggio di origine  $(P, k)$  e di colore  $\alpha$ .

- **specchio** $(x, y, w, z, a_1 \rightarrow \sigma_1, a_2 \rightarrow \sigma_2, \dots, a_n \rightarrow \sigma_n)$

Se  $(x, y)$  e  $(w, z)$  sono le coordinate dei vertici della diagonale di un quadrato, allora immette nel piano lo specchio  $S(x, y, w, z)$  e gli assegna le regole di ricolorazione  $a_1 \rightarrow \sigma_1, a_2 \rightarrow \sigma_2, \dots, a_n \rightarrow \sigma_n$ . Altrimenti non esegue alcuna operazione.

Vedi la sezione seguente per le specifiche di input.

- **elimina** $(x, y)$

Elimina dal piano tutti gli specchi contenenti il punto  $(x, y)$  (nota che  $(x, y)$  può anche giacere sul bordo quadrato di uno specchio).

- **punto** $(x, y, P, k, \alpha)$

Assumendo che il piano sia attraversato dal raggio di origine  $(P, k)$  e colore  $\alpha$ , stampa la stringa corrispondente al colore che assume il punto di coordinate  $(x, y)$  quando è attraversato per l'ultima volta da questo raggio. Se il punto  $(x, y)$  non è mai attraversato dal raggio stampa il messaggio

$(x, y)$  non attraversato dal raggio  $(P, k, \alpha)$

## 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  $x, y, w, z, k$  sono numeri interi relativi,  $P \in \{N, S, E, W\}$ ,  $a_1, \dots, a_n$  sono lettere minuscole, ossia elementi dell'alfabeto  $\mathcal{A}$ , e  $\sigma_1, \dots, \sigma_n$  sono stringhe finite, eventualmente vuote, sull'alfabeto  $\mathcal{A}$ . 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  |
|--|---|
| <code>r P k a<sub>1</sub>a<sub>2</sub>...a<sub>n</sub></code>  | <b>raggio</b> ( $P, k, a_1 a_2 \dots a_n$ )   |
| <code>s x y w z (a<sub>1</sub>, <math>\sigma_1</math>; a<sub>2</sub>, <math>\sigma_2</math>; ...; a<sub>n</sub>, <math>\sigma_n</math>)</code> | <b>specchio</b> ( $x, y, w, z, a_1 \rightarrow \sigma_1, a_2 \rightarrow \sigma_2, \dots, a_n \rightarrow \sigma_n$ ) |
| <code>e x y</code>   | <b>elimina</b> ( $x, y$ )   |
| <code>p x y P k a<sub>1</sub>a<sub>2</sub>...a<sub>n</sub></code>  | <b>punto</b> ( $x, y, P, k, a_1 a_2 \dots a_n$ )  |
| <code>q</code>   | Termina l'esecuzione del programma  |

Tabella 1: Specifiche del programma

È possibile che allo specchio  $S(x, y, w, z)$  non sia associata alcuna regola di ricolorazione. In tal caso tale specchio viene immesso con la riga di input:

```
s x y w z ()
```

### Note

1. Non devono essere presenti vincoli sulla dimensione del piano e sulla dimensione e numero degli specchi presenti nel piano. 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()`.

### Esempio

Si supponga che le linee di input siano:

```
s -5 -2 -2 -5 (a,ab)
s -5 2 -2 5 (a,ab)
s 3 6 7 2 (a,ab)
s 1 1 5 6 ()
s 7 -2 4 -5 (a,ab)
r W 5 a
s 3 -2 5 0 (a,ab)
```

```

s -3 0 0 -3 (a,ab;b,aca)
r W 5 a
s -2 -3 5 4 (a,bc)
r W 5 a
e 0 0
r W 5 a
s -7 1 -1 7 (a,aa;b,c)
s -9 1 -1 -7 (c,abc;a,)
s 2 9 8 3 ()
s 1 -9 9 -1 (c,ac)
r W 8 aa
r W 7 aa
p 5 5 W 7 c
p 3 2 W 7 zz
e -6 4
s -3 4 1 0 (a,aa;b,bb;c,cc)
r W 8 aa
r W 7 aa
p 5 5 W 7 c
r N -9 abc
r S 9 abc

```

L'output prodotto dal programma deve essere:

```

S 4 ab
N -2 abacaacaacaacaabacaacaacacabacaacaacaabacaacaacacabacaacaaca
N 6 bc
E 5 a
W -7 aa
E 3
aaccaacc
(3,2) non attraversato dal raggio (W, 7, zz)
S -8 aaaa
S -7 aaaa
(5,5) non attraversato dal raggio (W, 7, c)
S -9 abc
N 9 abc

```

## Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo [aguzzoli@dsi.unimi.it](mailto:aguzzoli@dsi.unimi.it) entro il 17 Giugno 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 17 Giugno 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.