Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Trubian, S. Aguzzoli

Appello dell'11 Gennaio 2005

Progetto "Controllo Remoto"
Consegna entro il 30 Gennaio 2005

Il problema

Obiettivo del progetto è studiare gli spostamenti di automi radiocomandati che si muovono su un terreno accidentato.

Formalmente, chiamiamo *piano* l'insieme dei punti

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

Ogni *automa* è identificato univocamente da un *nome* η, che è una stringa finita sull'alfabeto $\{0, 1\}$ ($\eta = b_1 b_2 \cdots b_n$ per qualche intero positivo n e $b_i \in \{0, 1\}$ per ogni $i \in \{1, \ldots, n\}$).

Lo stato di un automa η è completamente caratterizzato dal valore di due parametri, modificabili nel tempo, *posizione* $P(\eta)$ e *direzione di marcia* $D(\eta)$.

La *posizione* $P(\eta)$ di un automa η in un dato momento è specificata dando le *coordinate* $(x_0, y_0) \in \mathbb{Z} \times \mathbb{Z}$ della *cella* di lato unitario in cui η si trova in quel momento. Più precisamente, $P(\eta) = (x_0, y_0)$ significa che η si trova nella cella definita dall'insieme dei punti

$$\{ (x, y) \in \mathbb{R} \times \mathbb{R} \mid x_0 \le x \le x_0 + 1,\, y_0 \le y \le y_0 + 1\}.$$

Per semplicità si assume che ogni automa presente nel piano si trovi sempre all'interno di una siffatta cella, e su una cella può esservi più di un automa.

La *direzione di marcia* $D(\eta)$ è codificata da una lettera appartenente all'insieme $\{N, E, S, W\}$. Se $D(\eta) = N$ l'automa η è predisposto a muoversi verso nord, vale a dire verso la direzione crescente dell'asse delle ordinate. Analogamente, se $D(\eta) = E$ l'automa η è predisposto a muoversi verso est, vale a dire verso la direzione crescente dell'asse delle ascisse. $D(\eta) = S$ — verso sud — è la direzione *opposta* a N, così come W — verso ovest — è la direzione *opposta* a E (ovviamente N è a sua volta opposta a S e E a W).

Nel piano sono presenti degli *ostacoli* che limitano le possibilità di spostamento degli automi. Ogni ostacolo è rappresentato da un rettangolo $R(x_0, y_0, x_1, y_1)$, dove x_0, x_1, y_0, y_1 sono interi e

$$R(x_0, y_0, x_1, y_1) \;=\; \{(x, y) \in \mathbb{R} \times \mathbb{R} \;\mid\; x_0 \le x \le x_1,\, y_0 \le y \le y_1\}.$$

È possibile che gli ostacoli si sovrappongano; nessun automa potrà posizionarsi in una cella contenuta in qualche ostacolo. Gli automi si muovono nel piano seguendo delle *istruzioni* impartite da una sorgente remota, compatibilmente con gli ostacoli incontrati sul loro percorso.

In assenza di istruzioni, ogni automa η è fermo in $P(\eta)$, ma è pronto a muoversi in una delle quattro direzioni cardinali N, E, S, W, come sotto specificato.

La sorgente remota può emettere *segnali* rappresentati da stringhe finite del tipo

$$\alpha, \beta$$

dove α è una stringa finita sull'alfabeto $\{0, 1\}$, mentre β è una sequenza finita $C_1 m_1 C_2 m_2 \cdots C_k m_k$ di *istruzioni*, ognuna delle quali della forma Cm, dove $C \in \{N, E, S, W, A\}$ e m è un intero positivo.

All'atto dell'emissione del segnale ogni automa determina se tale segnale lo riguarda o meno: l'automa di nome η deve eseguire le istruzioni codificate nella stringa β del segnale se e solo se α è un prefisso di η, vale a dire $\alpha = b_1 b_2 \cdots b_h$ e $\eta = b_1 b_2 \cdots b_n$ per qualche $h \leq n$.

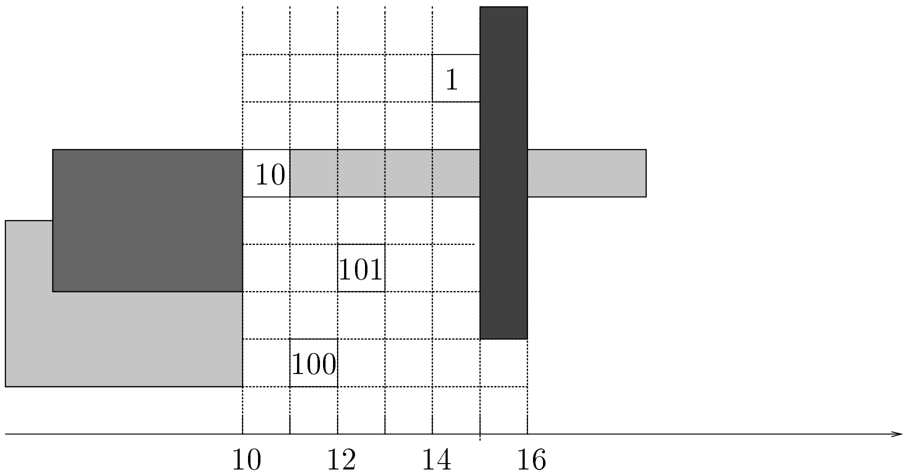
Le istruzioni vengono eseguite dagli automi nell'ordine in cui si susseguono in β. Supponiamo che un'automa η debba eseguire l'istruzione Cm. L'automa deve compiere m mosse, ciascuna delle quali può essere:

- uno *spostamento* in direzione C (se $C = A$, la direzione è $D(\eta)$);
- se lo spostamento non è possibile per la presenza di un ostacolo, un'*inversione* di direzione e la nuova direzione è la direzione opposta a quella attuale.

Al termine dell'istruzione, $P(\eta)$ è la posizione in cui si trova l'automa e $D(\eta)$ è l'ultima direzione considerata.

Esempio

Si supponga che nel piano ci siano gli automi 1, 10, 101 e 100 e quattro ostacoli come nella figura



dove

$$P(1) = (14, 7) \qquad P(10) = (10, 5) \qquad P(101) = (12, 3) \qquad P(100) = (11, 1)$$

e che il segnale emesso sia

$$1 \quad W10A5$$

Questo significa tutti gli automi presenti nel piano devono eseguire le istruzioni $W10$ e $A5$.

L'automa 1 compie 10 spostamenti in direzione W, quindi altri 5 spostamenti sempre in direzione W, e si ferma sulla posizione (-1,7).

L'automa 10 non può spostarsi, ma le uniche mosse che fa sono i cambi di direzione. Dopo la prima istruzione, la direzione corrente di 10 è W (ultima direzione considerata), dopo la seconda istruzione, la direzione corrente di 10 è E.

L'automa 101 può compiere 2 mosse in direzione W, quindi, essendo di fronte a un ostacolo, deve cambiare direzione e proseguire in direzione E; avanza di 4 caselle in direzione E fino ad arrivare in (14,3). A questo punto compie un'altra inversione di marcia e si sposta di 2 caselle verso W; al termine della prima istruzione $P(101) = (12, 3)$ e $D(101) = W$. Ora esegue $A5$, equivalente a $W5$, e di nuovo $P(101) = (12, 3)$, mentre $D(101) = E$.

Per quanto riguarda 100, dopo la prima istruzione $P(100) = (18, 1)$ e $D(100) = E$, dopo la seconda istruzione $P(100) = (23, 1)$ e $D(100) = E$.

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

- **crea** ()

  Crea un piano vuoto (eliminando l'eventuale piano già esistente).

- **automa** (x_0, y_0, C, η)

  Se la cella di coordinate (x_0, y_0) appartiene a qualche ostacolo, allora non esegue alcuna operazione. Altrimenti, se non esiste alcun automa di nome η lo crea; quindi riposiziona sul piano l'automa η nella cella di coordinate (x_0, y_0), e gli attribuisce la direzione di marcia $D(\eta) = C \in \{N, E, S, W\}$.

- **ostacolo** (x_0, y_0, x_1, y_1)

  Se le celle contenute nel rettangolo $R(x_0, y_0, x_1, y_1)$ non contengono alcun automa, inserisce l'ostacolo rappresentato da $R(x_0, y_0, x_1, y_1)$, altrimenti non compie alcuna operazione.

- **segnale** (α, β)

  Emette il segnale α, β.

- **posizione** α

  Restituisce la lista di tutti gli automi η tali che α è un prefisso di η, specificando l'attuale posizione $P(\eta)$ e l'attuale direzione di marcia $D(\eta)$, nel formato definito nelle note della sezione *Specifiche di implementazione*.

- **elimina** (x, y)

  Elimina gli eventuali ostacoli contenenenti il punto (x, y).

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 interi, $C \in \{N, E, S, W\}$, α è una stringa finita sull'alfabeto $\{0, 1\}$, β è una sequenza finita $C_1 m_1 C_2 m_2 \cdots C_k m_k$ dove $C_i \in \{N, E, S, W, A\}$ e $0 < m_i \in \mathbb{N}$, per ogni $i \in \{1, \ldots, k\}$.

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
`c`	**crea** ()
`a` $a \quad b \quad C \quad \alpha$	**automa** (a, b, C, α)
`o` $a \quad b \quad c \quad d$	**ostacolo** (a, b, c, d)
`s` $\alpha \quad \beta$	**segnale** (α, β)
`p` α	**posizione** (α)
`e` $a \quad b$	**elimina** (a, b)
`f`	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

3. Siano $\eta_1, \ldots, \eta_h$ tutti i nomi di automi nel piano di cui α è un prefisso, e sia per ogni $i \in \{1, \ldots, h\}$, $P(\eta_i) = (x_i, y_i)$ e $D(\eta_i) = C_i$. Allora l'output del comando `p` α deve essere visualizzato nel seguente formato:
   (
   $\eta_1 : x_1, y_1, C_1$
   $\eta_2 : x_2, y_2, C_2$
   $\vdots$
   $\eta_h : x_h, y_h, C_h$
)

Esempio
Si supponga che le linee di input siano:

```
c
a 12 6 S 001
a 12 3 N 010
a 15 6 W 011
a  9 6 E 0101
o -2 -1 7 8
o 3 -5 10 6
o 15 2 18 5
o 8 1 13 4
o 7 10 18 12
o 8 -4 19 -2
s 01 A3
p 0
o 15 -3 18 11
s 0 A4E7A8
p 0
```

```
e 12 1
e 16 2
a 16 0 W 001
a 16 0 S 0010
s 001 A10S4E1W2
p 001
p 011
f
```

L'output prodotto dal programma deve essere:

```
(
001: 12, 6, S
010: 12, 6, N
011: 12, 6, W
0101: 12, 6, E
)
(
001: 12, 2, W
010: 2, 9, W
011: 7, 6, E
0101: 12, 6, E
)
(
001: 12, -1, W
0010: 15, 1, W
)
(
011: 7, 6, E
)
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro il 30 Gennaio 2005 (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 30 Gennaio 2005 (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.