Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Goldwurm, S. Aguzzoli

Appello del 9 gennaio 2002

Progetto “MiniLife”
Consegna entro il 31 gennaio 2002

Il problema

Obiettivo del progetto è studiare l'evoluzione di un semplice automa cellulare bidimensionale.

L'universo è rappresentato da un piano di dimensione fissata $n \times n$ ($n > 0$ intero) suddiviso in quadrati di lato unitario che chiamiamo *celle*. L'*intorno* di una cella s del piano è l'insieme delle 8 celle che la circondano (con le ovvie limitazioni quando la cella in questione tocca qualche lato del piano). In un dato istante t, ogni cella può essere *viva* o *morta*.

Partendo da una situazione iniziale (al tempo $t = 0$), in cui viene dato un insieme V_0 di celle vive nel piano, a ogni istante successivo $t = 1, 2, ...$, l'insieme V_t delle celle vive al tempo t è determinato applicando all'insieme V_{t-1} la regola di evoluzione specificata nel seguito.

La *regola di evoluzione* di ogni cella s è la seguente:

- se s è viva al tempo $t - 1$ rimane viva al tempo t;
- se al tempo $t - 1$ s è morta e il suo intorno conta esattamente 3 celle vive, allora s diventa viva al tempo t (diciamo anche che s nasce);
- in nessun altro caso s può nascere (tranne che per inserzione manuale, vedi sotto).

Oltre alla regola di evoluzione si prevede di poter modificare lo stato delle celle mediante inserzione ed eliminazione manuale di celle vive.

Formalmente, fissato un intero n, chiamiamo *piano* l'insieme dei punti

$$\text{Piano}(n) \;=\; \{\, (x, y) \in \mathbf{N} \times \mathbf{N} \quad | \quad 0 \le x \le n,\ 0 \le y \le n\}.$$

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):

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

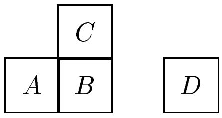
Ad esempio, Cella$(3, 5) \;=\; \{(3, 5), (3, 6), (4, 6), (4, 5)\}$

Analogamente, definiamo *Rettangolo*(a, b, c, d) come l'insieme dei punti del piano individuati dai punti (a, b) e (c, d) (coordinate dei vertici più in basso a sinistra e più in alto a destra), ossia

$$\text{Rettangolo}(a, b, c, d) \;=\; \{\, (x, y) \quad | \quad a \le x \le c,\ b \le y \le d\}.$$

Chiamiamo (a, b, c, d) le *coordinate* di Rettangolo(a,b,c,d).
Ad esempio, Rettangolo$(1, 3, 2, 5)$ è il rettangolo formato dai punti $\{(1, 3), (2, 3), (1, 4), (2, 4), (1, 5), (2, 5)\}$.

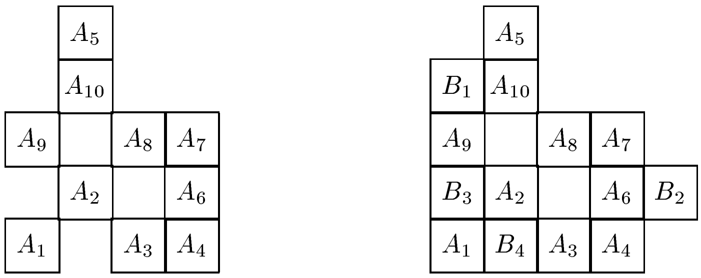
Diciamo che due celle sono *adiacenti* se hanno in comune almeno un punto (quindi, ogni cella è adiacente a se stessa); nella figura qui sotto le celle A e B sono adiacenti, B e C sono adiacenti, e anche A e C sono adiacenti. Al contrario D non è adiacente ad alcuna delle altre tre celle.



Un *cammino* da A_1 a A_h è una sequenza di celle $A_1, \ldots, A_h$ 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 vive 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 vive $\mathcal{A}$ tale che $\mathcal{A}$ è connesso e, per ogni cella viva $P \notin \mathcal{A}$, $\mathcal{A} \cup \{P\}$ non è connesso. La *dimensione* di un blocco è data dal numero di celle vive che lo compongono. L'*intorno* della cella $\mathcal{A}$ (viva o morta) è l'insieme delle celle adiacenti ad $\mathcal{A}$ tranne $\mathcal{A}$ stessa.

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 $\{D\}$.

Si consideri l'esempio seguente:



Nella figura a sinistra è riportata la situazione al tempo t. Le celle etichettate con la lettera A sono tutte e sole le celle vive al tempo t. Nella figura a destra la situazione al tempo $t+1$ ottenuta dopo un passo di evoluzione. Le celle etichettate con la lettera B sono tutte e sole quelle nate al tempo $t+1$. Le celle vive al tempo $t+1$ sono tutte e sole le celle etichettate con la lettera A o con la lettera B.

Si richiede di implementare una struttura dati efficiente che permette di eseguire le operazioni seguenti (si tenga presente che la dimensione del piano n può essere grande rispetto al numero di celle vive, quindi *non è sicuramente efficiente rappresentare il piano mediante una bit-map di tipo* $n \times n$).

Nel seguito parleremo di **inserimento** di Cella(x,y) per riferirci all'operazione che pone Cella(x,y) nello stato *viva*.
Analogamente chiameremo **eliminazione** di Cella(x,y) l'operazione che pone Cella(x,y) nello stato *morta*.

- **crea (n)**

  Crea un piano vuoto di dimensione **n** (se un piano è già definito viene eliminato).

- **inserisci (x, y)**

  Inserisce nel piano la Cella(**x, y**).

- **inserisci-rand (k)**

  Inserisce casualmente **k** nuove celle vive nel piano.

- **genera ()**

  Assumendo di essere al tempo t, genera le celle che devono nascere al tempo $t+1$. Questo determina un passo (da t a $t+1$) nell'evoluzione dell'intero sistema.

- **elimina (x, y)**

  Elimina dal piano la Cella(**x, y**).

- **listacelle ()**

  Fornisce la lista di tutte le celle vive al tempo corrente, secondo le specifiche sotto riportate.

- **min-rett (x, y)**

  Calcola le coordinate del più piccolo rettangolo contenente il blocco cui appartiene Cella(x,y). se Cella(x,y) non è viva restituisce il rettangolo degenere Rettangolo(x,y,x,y).

- **num-blocchi ()**

  Calcola il numero di blocchi presenti nel piano.

- **max-dim-blocco ()**

  Calcola la dimensione del più grande blocco presente nel piano.

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, z, w sono numeri naturali 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.

Linea di input	Operazione
`c` x	**crea (x)**
`i` x y	**inserisci (x, y)**
`d` x	**inserisci-rand (x)**
`g`	Calcola la funzione **genera()**
`e` x y	**elimina (x, y)**
`l`	Stampa la lista delle celle vive secondo il formato sotto specificato
`r` x y	Stampa le coordinate del minimo rettangolo calcolato da **min-rett(x, y)** secondo il formato sotto specificato
`b`	Stampa il valore di **num-blocchi ()**
`m`	Stampa il valore di **max-dim-blocco ()**
`f`	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

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()`.

Formato per la visualizzazione della lista delle celle vive
Il formato a cui attenersi è il seguente: Assumiamo che l'insieme delle celle vive sia

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

dove, per ogni $i \in \{1, \ldots, n-1\}$, valga $(x_i, y_i) < (x_{i+1}, y_{i+1})$ rispetto all' ordine stretto $<$ definito da:

$$(x_i, y_i) < (x_{i+1}, y_{i+1}) \quad \text{se e solo se} \quad x_i < x_{i+1} \quad \text{oppure} \quad x_i = x_{i+1} \text{ e } y_i < y_{i+1}.$$

L'output della funzione **listacelle()** , innescata dal comando **l**, deve essere:

```
x1 y1
x2 y2
..
xn yn
```

Formato per la visualizzazione delle coordinate del minimo rettangolo
Sia Rettangolo(a,b,c,d) il minimo rettangolo che contiene il blocco a cui appartiene Cella(x,y).
L'output di min-rett(x,y) deve essere:

```
a b c d
```

Esempio. Si supponga che le linee di input siano:

```
c 100
i 3 2
i 4 2
i 5 2
i 2 5
i 3 5
i 4 5
b
g
l
b
r 4 2
e 5 2
e 3 4
e 4 3
e 3 5
g
l
b
m
f
```

L'output prodotto dal programma deve essere:

```
2
2 5
3 2
3 4
3 5
3 6
4 1
4 2
4 3
4 5
5 2
1
2 1 6 7
2 5
3 1
3 2
3 5
3 6
4 1
4 2
4 5
2
4
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica al Dr. Stefano Aguzzoli all'indirizzo aguzzoli@dsi.unimi.it entro il 31 gennaio 2002. L'esame orale si svolgerà martedì 5 febbraio alle ore 9 in aula 6 presso il Dip. di Scienze dell'Informazione in via Comelico 39/41.

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.

I due file suddetti devono essere contenuti in un unico file .zip il cui nome dovrà essere cognome.zip.
La relazione e il codice dovranno riportare il vostro nome, cognome e matricola.
Una copia cartacea della relazione e del listato deve inoltre essere consegnata al Dr. Aguzzoli sempre entro il 31 gennaio 2002 (lasciandola eventualmente nella sua casella postale presso il dipartimento in via Comelico).

Si ricorda infine di presentarsi alla prova orale con la stampa cartacea della relazione e del listato.

Per ogni ulteriore chiarimento:
E-mail: aguzzoli@dsi.unimi.it
Ricevimento: Mercoledì, h. 15-16, stanza S211.