

# Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Torelli, S. Aguzzoli

Appello dell'8 gennaio 2009

Progetto "Edilizia"

Consegna entro il 29 gennaio 2009

## Il problema

In una città in forte espansione si stanno erigendo nuovi quartieri e nuove strade per collegarli. Formalmente, rappresentiamo un *quartiere*  $Q$  in costruzione come un insieme di edifici:

$$Q = \{E_1, \dots, E_u\}.$$

Un *edificio*  $E = E(l, r, h)$  è un rettangolo, specificato dalla sua *base*  $(l, r) \in \mathbb{Z}^2$ ,  $l < r$ , e dalla sua *altezza*  $h \in \mathbb{Z}$ ,  $h > 0$ .

$$E = \{(x, y) \in \mathbb{Z}^2 \mid l \leq x \leq r, 0 \leq y \leq h\}.$$

Il *profilo*  $P(Q)$  di un quartiere  $Q = \{E_1, \dots, E_u\}$  è l'unione di tutti gli edifici che lo compongono:

$$P(Q) = \bigcup_{i=1}^u E_i.$$

Si noti che la *frontiera* dell'insieme  $P(Q)$  è una linea spezzata formata da segmenti alternatamente orizzontali e verticali. L'intersezione di ogni coppia di segmenti consecutivi della frontiera di  $P(Q)$  è un punto  $(x, y) \in \mathbb{Z}^2$ . Chiamiamo ogni siffatto punto, *vertice* di  $P(Q)$ . Per specificare un profilo basta fornire la sequenza dei vertici del profilo stesso in senso orario, partendo dal vertice con ascissa minima e ordinata = 0.

### Esempio 1

Si supponga che il quartiere  $Q$  contenga i seguenti edifici:  $E_1 = E(-4, -2, 7)$ ,  $E_2 = E(-3, 2, 3)$ ,  $E_3 = E(-3, 3, 4)$ ,  $E_4 = E(2, 5, 6)$ ,  $E_5 = E(6, 8, 5)$ .

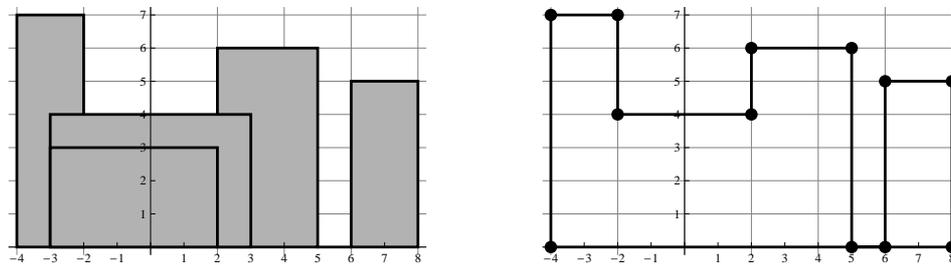


Figura 1: Gli edifici  $E_1, E_2, E_3, E_4, E_5$  dell'Esempio 1, il profilo del quartiere che determinano, e i vertici del profilo stesso.

Il profilo  $P(Q)$  è specificato dalla sequenza

$$(-4, 0), (-4, 7), (-2, 7), (-2, 4), (2, 4), (2, 6), (5, 6), (5, 0), (6, 0), (6, 5), (8, 5), (8, 0).$$

Ogni quartiere  $Q$  ha un nome  $N(Q)$ , che è una stringa di lunghezza arbitraria sull'alfabeto  $\{a, b, \dots, z\}$  delle lettere minuscole.

Una *strada*  $S = S(Q_1, Q_2)$  collega due quartieri  $Q_1$  e  $Q_2$  (non necessariamente distinti) ed è caratterizzata da una lunghezza  $L(S)$ , dove  $L(S) \in \mathbb{Z}$ ,  $L(S) > 0$ . (Ogni strada è percorribile in entrambi i sensi). Due quartieri possono essere collegati da più strade distinte.

Un *percorso*  $\pi = (S_1, S_2, \dots, S_v)$  è una sequenza di strade tale che, o  $v = 1$  (il percorso è costituito da una sola strada), o per ogni  $i = 1, 2, \dots, v - 1$  esistono tre quartieri  $Q_{i1}, Q_{i2}, Q_{i3}$ , non necessariamente distinti, tali che  $S_i$  collega  $Q_{i1}$  a  $Q_{i2}$  e  $S_{i+1}$  collega  $Q_{i2}$  a  $Q_{i3}$ .

La *lunghezza*  $l(\pi)$  di un percorso  $\pi$  è la somma delle lunghezze delle strade che lo compongono:

$$l(\pi) = \sum_{i=1}^v L(S_i).$$

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 edifici di un quartiere può essere molto grande rispetto al numero di edifici presenti, quindi *non è sicuramente efficiente rappresentare l'insieme degli edifici di un quartiere mediante un'unica matrice* contenente i punti appartenenti al profilo del quartiere stesso).

- **edificio** (*nome, l, r, h*)

Inserisce un edificio di base  $(l, r)$  ed altezza  $h$  nel quartiere di nome *nome*.

- **strada** (*nome1, nome2, l*)

Costruisce una strada di lunghezza  $l$  che collega il quartiere di nome *nome1* al quartiere di nome *nome2*.

- **profilo** (*nome*)

Visualizza il profilo del quartiere di nome *nome*, secondo le specifiche riportate nell'apposita sezione. Se tale quartiere non contiene edifici, non visualizza nulla.

- **distanza** (*nome1, nome2*)

Calcola e stampa la lunghezza minima di un percorso tra il quartiere *nome1* e il quartiere *nome2*. Se non esiste un percorso tra questi due quartieri stampa il messaggio:  
*nome2 non e' raggiungibile da nome1.*

- **collegamenti** (*nome*)

Stampa i nomi di tutti i quartieri collegati al quartiere di nome *nome* tramite una strada, secondo le specifiche riportate nell'apposita sezione.

Si noti che le operazioni richieste sono liberamente implementabili; in particolare, non vanno necessariamente intese come prototipi di funzioni.

## Specifiche di implementazione

Il programma deve leggere dallo standard input (`stdin`) una sequenza di righe (separate da `\n`), ciascuna delle quali corrisponde a una riga della prima colonna della Tabella 1, dove,  $\alpha$  e  $\beta$  sono stringhe finite sull'alfabeto  $\{a, b, \dots, z\}$  di lunghezza *arbitraria* (ossia, non deve essere definita nel programma alcuna costante  $k$  che limiti la lunghezza di una stringa a  $k$ );  $a$  e  $b$  sono interi e  $c$  è un intero positivo. I vari elementi sulla riga sono separati da uno o più spazi. Quando una riga 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 riga.

RIGA DI INPUT	OPERAZIONE
<code>e <math>\alpha</math> a b c</code>	<b>edificio</b> ( $\alpha, a, b, c$ )
<code>s <math>\alpha</math> <math>\beta</math> c</code>	<b>strada</b> ( $\alpha, \beta, c$ )
<code>p <math>\alpha</math></code>	<b>profilo</b> ( $\alpha$ )
<code>d <math>\alpha</math> <math>\beta</math></code>	<b>distanza</b> ( $\alpha, \beta$ )
<code>c <math>\alpha</math></code>	<b>collegamenti</b> ( $\alpha$ )
<code>f</code>	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

### Note

1. Non devono essere presenti vincoli sul numero di quartieri e di edifici, sulle loro dimensioni e collocazione, sul numero di strade e sulle loro lunghezze (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()`.

### 3. specifiche per la visualizzazione di un profilo

Sia  $P(Q)$  il profilo del quartiere di nome *nome* e siano  $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$  i vertici del profilo stesso, elencati in ordine orario, a partire dal punto  $(x_1, y_1)$  tale che  $x_1 = \min_{i=1}^k \{x_i\}$  e  $y_1 = 0$ . Allora l'output del comando

```
p nome
```

deve essere visualizzato come:

```
(  
x1, y1  
x2, y2  
⋮
```

$x_k, y_k$   
)

#### 4. specifiche per la visualizzazione dei collegamenti di un quartiere

Sia  $Q$  il quartiere di nome  $nome$  e sia  $\{Q_1, Q_2, \dots, Q_k\}$  l'insieme dei quartieri collegati a  $Q$  tramite una strada. Si noti che  $Q$  potrebbe appartenere a questo insieme. Per ogni  $i = 1, 2, \dots, k$  sia  $nome_i = N(Q_i)$ . Allora l'output del comando

`c nome`

deve essere visualizzato come:

```
(  
nome1  
nome2  
:  
nomek  
)
```

Non è rilevante l'ordine in cui appaiono i nomi.

#### Esempio

Si supponga che le righe di input siano:

```
e centro 2 5 5  
s centro nord 15  
s centro ovest 7  
s ovest centro 10  
s ovest ovest 4  
e verde 7 10 4  
e rosso -2 2 6  
e giallo 0 10 1  
e centro 0 8 2  
e centro 4 7 3  
p centro  
c centro  
c ovest  
s ovest nord 4  
s centro est 2  
s ovest sud 6  
s est sud 2  
d centro verde  
d centro nord  
d nord est  
s rosso verde 4  
s rosso giallo 2  
s verde giallo 7  
d giallo verde  
s verde giallo 5  
c giallo
```

```
s giallo ovest 3
d nord verde
e giallo -20 -1 1
p giallo
e giallo -2 2 4
p giallo
d verde blu
e centro 1 6 4
e centro 4 6 10
p centro
e centro 7 10 2
p centro
s est rosso 7
d giallo centro
d centro verde
e centro 0 10 10
p centro
f
```

L'output prodotto dal programma deve essere il seguente

```
(
0,0
0,2
2,2
2,5
5,5
5,3
7,3
7,2
8,2
8,0
)
(
ovest
nord
)
(
ovest
centro
)
verde non e' raggiungibile da centro
11
12
6
(
verde
rosso
)
12
```

(  
-20,0  
-20,1  
-1,1  
-1,0  
0,0  
0,1  
10,1  
10,0  
)

(  
-20,0  
-20,1  
-2,1  
-2,4  
2,4  
2,1  
10,1  
10,0  
)

blu non e' raggiungibile da verde

(  
0,0  
0,2  
1,2  
1,4  
2,4  
2,5  
4,5  
4,10  
6,10  
6,3  
7,3  
7,2  
8,2  
8,0  
)

(  
0,0  
0,2  
1,2  
1,4  
2,4  
2,5  
4,5  
4,10  
6,10  
6,3  
7,3  
)

7,2  
10,2  
10,0  
)  
10  
13  
(  
0,0  
0,10  
10,10  
10,0  
)

## Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo [aguzzoli@dsi.unimi.it](mailto:aguzzoli@dsi.unimi.it) entro il 29 gennaio 2009 (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/~aguzzoli/algo.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 entro il 29 gennaio 2009 (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.

Il calendario degli esami orali sarà disponibile sulla pagina del corso qualche giorno dopo il termine di consegna del progetto.

Per ogni ulteriore chiarimento:

E-mail: [aguzzoli@dsi.unimi.it](mailto: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.

La realizzazione del progetto è una prova d'esame da svolgersi **individualmente**. I progetti giudicati frutto di **collaborazioni** saranno **estromessi** d'ufficio dalla valutazione.