

Laboratorio di Algoritmi e Strutture Dati

Docente: S. Aguzzoli

Progetto “Isole”

valido per l'appello di gennaio 2013

Il problema

Le isole di un arcipelago sono collegate da due tipi di ponti: ponti carrabili dove possono passare auto e pedoni pagando un pedaggio, e ponti solo pedonali che invece sono di uso gratuito.

I ponti pedonali però sono piccoli e più fragili di quelli carrabili: spesso le mareggiate ne distruggono alcuni.

Un **isola** è identificata da un *nome*.

Un **ponte carrabile** è invece identificato dalla *coppia dei nomi* delle isole che congiunge, e da un intero compreso tra 0 e 50 che determina il prezzo da pagare per attraversarlo (il *pedaggio*).

Un **ponte pedonale** è identificato solo dalla *coppia dei nomi* delle isole che congiunge.

Le auto possono attraversare solo i ponti carrabili, mentre i pedoni posso utilizzare sia i ponti carrabili che quelli pedonali, ma quando usano i ponti carrabili pagano lo stesso pedaggio delle auto.

Per andare da un'isola A ad un'isola B si può attraversare un ponte che le collega, se esiste, pagando il pedaggio relativo a quel ponte se il ponte è carrabile.

Oppure se esistono le isole A_1, \dots, A_n tali che A è collegata con un ponte a A_1 , ogni A_i è collegata ad A_{i+1} , e A_n è collegata a B , si può andare da A a B attraversando questi $n + 1$ ponti: in questo caso il pedaggio da pagare è la somma degli eventuali pedaggi degli $n + 1$ ponti attraversati. Nota che per andare da un'isola ad un'altra in genere sono possibili diversi percorsi, corrispondenti a diversi pedaggi. In alcuni casi non ci sono percorsi possibili e le due isole non sono raggiungibili una a partire dall'altra.

Esempio. Supponiamo che nell'arcipelago ci siano le isole A, B, C, D ed E . Inoltre supponiamo che esistano i seguenti ponti:

- tra A e B c'è un ponte pedonale;
- tra A e E c'è un ponte carrabile con pedaggio 10 e un ponte pedonale;
- tra C e E c'è un ponte carrabile con pedaggio 7;
- tra D e B c'è un ponte pedonale;
- tra C e D c'è un ponte carrabile con pedaggio 23 e un ponte pedonale.

Allora un pedone che deve andare da A a C , può fare diverse strade: può andare prima sull'isola E tramite il ponte pedonale e poi su C usando il ponte carrabile e quindi pagando un pedaggio di 7; oppure può usare solo ponti pedonali andando da A a B , quindi da B a D e infine da D a C . Un'auto che vuole andare da A a C deve invece attraversare necessariamente i ponti carrabili da A ad E (pagando 10) e poi da E a C (pagando 7), e quindi pagherà un totale di 17.

Una **mareggiata** distrugge casualmente alcuni dei ponti pedonali. Nell'esempio precedente, una mareggiata potrebbe distruggere il ponte pedonale tra A e B , quindi un pedone che voglia andare da A a B sarai costretto a passare da E , poi da C , quindi da D ed infine arrivare a B .

Si richiede di implementare una struttura dati efficiente che permetta di eseguire le operazioni seguenti.

- **nuovo** ()
Elimina tutte le isole e i ponti esistenti.
- **crea-pontepedonale** ($w1, w2$)
Crea un ponte pedonale tra l'isola di nome $w1$ e l'isola di nome $w2$. Se le isole sono già collegate da un ponte pedonale allora non compie alcuna operazione.
- **crea-pontecarrabile** ($w1, w2, p$)
Crea un ponte carrabile tra l'isola di nome $w1$ e l'isola di nome $w2$, con pedaggio p . Se le isole sono già collegate da un ponte carrabile (anche se con diverso pedaggio) allora non compie alcuna operazione.
- **stampa-ponti** ()
Stampa tutti i ponti presenti (sia pedonali che carrabili) secondo il formato descritto nell'apposita sezione.
- **mareggiata** ()
Distrugge un numero casuale di ponti pedonali (eliminandoli dalla struttura) e restituisce l'elenco dei ponti distrutti.
- **collega-pedonale** ($w1, w2$)
Trova un collegamento pedonale tra le isole di nome $w1$ e $w2$ in modo che la somma dei pedaggi dei ponti attraversati sia la più piccola possibile (eventualmente uguale a zero). Se non è possibile andare dall'isola $w1$ all'isola $w2$ attraversando dei ponti allora restituisce un messaggio di errore.
- **collega-carrabile** ($w1, w2$)
Trova un collegamento carrabile tra le isole di nome $w1$ e $w2$ in modo che la somma dei pedaggi dei ponti attraversati sia la più piccola possibile. Se non è possibile andare dall'isola $w1$ all'isola $w2$ attraversando dei ponti carrabili allora restituisce un messaggio di errore.

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 u, v sono stringhe non vuote e n è un numero intero positivo minore o uguale a 50

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. I vari elementi sulla riga sono separati da uno o più spazi.

Output di s. Per ogni isola A , il comando `s` deve restituire l'elenco delle isole che sono collegate ad A tramite ponti pedonali e l'elenco delle isole che sono collegate con ponti carrabili con il relativo pedaggio tra parentesi quadre. Nell'esempio precedente l'output di `s` è:

```
A {pedonale: B, E; carrabile: E [10]}
B {pedonale: A, D;}
C {pedonale: D; carrabile: D [23], E [7]}
D {pedonale: B, C; carrabile: C [23]}
E {pedonale: A; carrabile: A [10], C [7]}
```

Se non ci sono ponti allora il comando `s` deve produrre come output la linea
non ci sono ponti adatti.

RIGA DI INPUT	OPERAZIONE
n	nuovo()
p u v	crea-pontepedonale(u, v)
P u v n	crea-pontecarrabile(u, v, n)
s	stampa-ponti()
c u v	collega-pedonale(u, v)
C u v	collega-carrabile(u, v)
m	mareggiata(u, v)
f	Termina l'esecuzione del programma

Tabella 1: Specifiche del programma

Note

1. Per leggere l'input si usino le funzioni standard ANSI C `getchar()` e/o `scanf()`.
2. Non devono essere presenti vincoli sul numero di ponti e di isole da inserire. 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()`.

Esempio Le righe che iniziano con `>` sono da intendersi come linee di input.

```
>p isolapiccola pescatori
>p isolabella isolapiccola
>p isolabella isoletta
>P isolapiccola grande 3
>P grande pescatori 7
>P pescatori isoletta 20
>P isoletta grande 5
>s
grande {carrabile: isolabella [3], isoletta [5], pescatori [7]}
isolabella {pedonale: isolapiccola, isoletta;}
isolapiccola {pedonale: isolabella, pescatori; carrabile: grande [3]}
isoletta {pedonale: isolabella; carrabile: grande [5], pescatori [20]}
pescatori {pedonale: isolapiccola; carrabile: grande [7], isoletta [20]}
>c isoletta pescatori
isoletta -> isolabella -> isolapiccola -> pescatori
pedaggio = 0
>C isoletta pescatori
isoletta -> grande -> pescatori
pedaggio = 12
>c grande isolabella
grande -> isolapiccola -> isolabella
pedaggio = 3
```

```
>C grande isolabella
non ci sono ponti adatti
>m
distrutti i ponti pedonali
isolabella isolapiccola
pescatori isolapiccola
>c grande isolabella
grande -> isoletta -> isolabella
pedaggio = 5
>f
```

1 Modalità di consegna

Il presente progetto è valido per l'appello del 10 gennaio 2013. Il progetto va inviato per posta elettronica all'indirizzo aguzzoli@dsi.unimi.it entro il 10 gennaio 2013. 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 le scelte implementative, analizzando il costo delle diverse operazioni richieste dalla specifica;

Tutti i file (file sorgenti + relazione) devono essere contenuti in un unico file **.zip** il cui nome dovrà essere della forma **cognome_matricola.zip**. La relazione e il codice devono riportare nome, cognome e matricola. Una copia cartacea della relazione e del codice deve inoltre essere consegnata al docente entro le scadenze fissate (lasciandola eventualmente nella sua casella postale presso il dipartimento in via Comelico).

La discussione dei progetti si svolgerà in data e luogo da specificarsi, Il calendario dei colloqui sarà disponibile sulla pagina del corso <http://homes.dsi.unimi.it/~aguzzoli/algo.html> qualche giorno dopo il termine di consegna del progetto.

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

Si richiede allo studente di effettuare un **adeguato collaudo** del proprio progetto su numerosi esempi diversi per verificarne la correttezza.

La versione aggiornata del progetto è pubblicata in **.pdf** sul sito:

<http://homes.dsi.unimi.it/~aguzzoli/algo.html>.

Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto.