

# Labirinti

## Progetto d'esame del corso di “Programmazione e Laboratorio di programmazione” Corso di Laurea in Informatica per le Telecomunicazioni

Appello del 26 settembre 2008

### 1 Introduzione

Scopo del progetto è realizzare un solutore per labirinti. Il programma deve leggere la configurazione del labirinto da standard input, calcolarne una soluzione e visualizzarla su standard output. Il programma deve essere scritto in ANSI C, rispettando rigorosamente le specifiche di formato per l'input e l'output. Il codice sorgente deve essere accompagnato da una relazione sintetica che illustri le caratteristiche del programma e le scelte fatte nelle fasi di progettazione e implementazione. Il progetto va svolto individualmente.

In questo documento sono formalizzati i concetti di base relativi ai labirinti perfetti, definite le specifiche di formato per l'input e l'output, precisate le modalità di consegna del progetto.

### 2 Definizioni preliminari

In enigmistica, per *labirinto* si intende un gioco di pazienza che consiste nell'individuare, in un disegno rappresentante un groviglio di *strade* e *pareti*, l'unico sentiero che conduce dall'*entrata* all'*uscita*. In questo contesto particolare, considereremo solo labirinti perfetti di forma rettangolare in cui le strade e gli ostacoli sono disposti parallelamente ai lati del bordo, e sia l'entrata che l'uscita sono collocati sul bordo stesso, rispettivamente nelle pareti di sinistra e di destra.



Figura 1: Un esempio di labirinto (a sinistra) con due possibili soluzioni (al centro e a destra).

Formalmente, rappresentiamo i labirinti attraverso matrici rettangolari di celle che possono essere *nere* o *bianche*: le celle nere rappresentano le pareti, quelle bianche le strade; tutte le celle del bordo saranno quindi piene, tranne che per due celle vuote che rappresentano l'entrata e l'uscita del labirinto. In questa rappresentazione, chiamiamo *sentiero* una qualunque sequenza di celle vuote in cui due coppie di celle consecutive sono sempre adiacenti (ovvero sono una sopra, o a fianco, dell'altra). Per *soluzione del labirinto* intendiamo

infine un sentiero che collega l'entrata e l'uscita del labirinto stesso. Graficamente, rappresentiamo le celle che compongono una soluzione come celle *grigie*.

In Figura 1 sono mostrati un labirinto e due sue possibili soluzioni. Si noti come la soluzione di destra si ottiene ripercorrendo alcune strade in entrambe le direzioni: questo succede quando si imbocca una strada che si dimostra essere un vicolo cieco ed è necessario tornare sui propri passi.

Una tecnica efficace per ottenere la soluzione di certi labirinti è la regola della mano destra. L'idea è molto semplice: posti nell'entrata del labirinto, è sufficiente camminare tenendo sempre la mano destra appoggiata alla parete, fino al raggiungimento dell'uscita. La soluzione di destra in Figura 1 è ottenuta applicando proprio questa regola. Sebbene tale tecnica non funzioni per ogni possibile labirinto, c'è un insieme di labirinti, cosiddetti *perfetti*, per cui viceversa essa produce sempre una soluzione.

### 3 Il formato PGM

Esistono molti formati per rappresentare immagini in toni di grigio. Il formato *PGM* (acronimo di Portable GrayMap) è uno di questi: è stato progettato per essere estremamente facile da capire e manipolare; esistono inoltre molte applicazioni in grado di visualizzare immagini nel formato PGM o di convertirle in altri formati (es: GIMP, ImageMagick). La specifica formale del formato può essere reperita a questo indirizzo:

<http://netpbm.sourceforge.net/doc/pgm.html> .

Qui ci limitiamo a illustrare il formato attraverso un esempio. L'immagine centrale di Figura 1 è un ingrandimento dell'immagine PGM seguente:

---

```
P2
17 11
2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 2 2 2 2 2 2 0 2 2 2 2 2 0
0 0 0 1 0 2 0 0 0 2 0 0 0 2 0 0 0
0 2 0 1 0 2 0 2 0 2 2 2 2 2 0 2 0
0 2 0 1 0 2 0 2 0 2 0 0 0 2 0 2 0
0 1 1 1 0 2 0 2 2 2 0 2 2 2 2 2 0
0 1 0 0 0 0 0 0 0 0 0 2 0 2 0 2 0
0 1 1 1 1 1 2 2 2 2 0 2 0 2 0 2 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

---

Vediamo di interpretare la codifica PGM. Il codice `P2` nella prima riga identifica il tipo di formato; i due numeri nella seconda riga rappresentano la larghezza in pixel (in questo caso 17) e l'altezza (in questo caso 11) dell'immagine; il numero nella terza riga (in questo caso 2) rappresenta il massimo tono di grigio utilizzato. Le ultime 11 righe rappresentano ciascuna una riga di pixel dell'immagine: per ogni pixel è indicato il suo tono di grigio con un valore compreso tra 0 e 2. Si può notare chiaramente che al valore 0 corrispondano le pareti del labirinto (nere), al valore 2 le strade (bianche), al valore 1 la soluzione (grigia).

### 4 Specifiche e suggerimenti

Il vostro programma dovrà ricevere da standard input un labirinto rettangolare (secondo la definizione data sopra) rappresentato in formato PGM, calcolarne una soluzione e stamparla su standard output nello stesso formato. Il programma non dovrà produrre altro output oltre alla soluzione.

Per le finalità di questo progetto, potete limitarvi a considerare solo labirinti perfetti e, quindi, ad implementare la regola della mano destra. Facoltativamente, è possibile anche studiare e implementare anche

altre strategie risolutive, o ampliare la classe dei possibili labirinti di input; in questo caso è importante documentare chiaramente le scelte fatte nella relazione che accompagna il progetto.

Prima e durante lo sviluppo del vostro elaborato, vi invito a prendere in considerazione i seguenti suggerimenti:

- valutate attentamente in che modo gestire le immagini PGM in input: il modo più semplice prevede l'utilizzo di un array bidimensionale, ma naturalmente questo comporta delle limitazioni sulle sue dimensioni e un uso non ottimale della memoria, in quanto la dimensione del labirinto da processare non è nota in fase di compilazione;
- è utile strutturare il programma usando delle funzioni, eventualmente raggruppandole in file distinti (es: funzioni di input/output, funzioni di calcolo della soluzione, ecc);
- al fine di rendere più leggibile il codice, usate delle macro (`#define`), ad esempio per definire pareti, strade, porzioni di soluzione.

Per testare il vostro programma, potete utilizzare l'archivio di labirinti perfetti e il loro generatore resi disponibile all'indirizzo

<http://lonati.dsi.unimi.it/progtelecom/labirinti.tar.gz>

Le istruzioni per l'utilizzo di tale generatore sono contenute nel file `README` dello stesso archivio.

## 5 Correzione e criteri di valutazione

La valutazione dell'elaborato dipenderà da molti fattori, tra cui:

- correttezza della soluzione,
- rispetto delle specifiche di formato,
- qualità dell'implementazione (principi di buona programmazione, tipi di dati usati, occupazione della memoria, ecc),
- stile del codice (indentazione, commenti, leggibilità, ecc),
- chiarezza, completezza e correttezza della relazione.

Naturalmente tali criteri si applicheranno anche alle parti facoltative eventualmente svolte.

## 6 Modalità di consegna

**Il progetto deve essere svolto individualmente.**

**La consegna del progetto deve avvenire tramite l'invio di una sola mail**, da spedire entro il giorno 23 settembre 2008 (incluso) all'indirizzo `lonati@dsi.unimi.it`. La mail deve tassativamente provenire dall'indirizzo di posta ufficiale dello studente, ossia l'indirizzo che termina con `@studenti.unimi.it`. La mail deve contenere un unico file allegato, di tipo archivio `zip`, il cui nome deve essere composto dal cognome seguito dal numero di matricola del studente (es: per lo studente Mario Rossi matr. 422481, il file deve chiamarsi `rossi422481.zip`); l'archivio deve contenere:

- il codice sorgente (compresi eventuali header file),
- la relazione in formato pdf o txt.

I sorgenti devono essere ANSI C e compilare tutti senza errori col compilatore `gcc` installato in laboratorio; non è consentito l'uso di librerie oltre alla *standard library*. Le specifiche di formato per l'input e l'output definite sopra devono essere rispettate rigorosamente.

**In presenza di errori di compilazione, la consegna non sarà ritenuta valida. Programmi consegnati diversamente da come specificato in questa sezione non saranno presi in considerazione.**

La discussione del progetto si svolgerà indicativamente il giorno 29 settembre insieme alla prova orale; orario e aula verranno specificati sul sito

<http://lonati.dsi.unimi.it/progtelecom/>.

**E' necessario presentarsi alla prova orale con una copia stampata della relazione e del codice.**