

Laboratorio di Algoritmi

Progetto “Traslochi” (gennaio 2025)

Nota: La scadenza del progetto è fissata per martedì 14 gennaio **compreso**.

Nota: Si consiglia di verificare ogni tanto se questo documento è stato aggiornato per correggere errori e rispondere a domande di altri studenti.

Il problema

Si vuole simulare un ambiente virtuale, costituito da un insieme di camere e corridoi distribuiti in un solo piano, in modo da preparare la realizzazione di un videogioco che si svolgerà in questo ambiente. È data la mappa dell’ambiente, nella quale sono marcate le posizioni di un agente mobile e di alcuni oggetti. Per semplicità, la mappa è rappresentata da una griglia rettangolare di punti, ciascuno indicato da una coppia di coordinate intere. La prima coordinata rappresenta l’indice di riga, misurato dall’alto verso il basso, la seconda l’indice di colonna, misurato da sinistra verso destra. Due punti si considerano adiacenti se uno dei due indici corrispondenti ha lo stesso valore e l’altro differisce per una sola unità (ad esempio, il punto $(2, 3)$ è adiacente ai punti $(3, 3)$ e $(2, 4)$, ma non al punto $(1, 2)$). Ogni punto ha quindi quattro posizioni adiacenti, due con lo stesso indice di riga e due con lo stesso indice di colonna, salvo i punti sui bordi o sugli angoli dell’ambiente, che ne hanno di meno. Ogni punto che appartiene all’ambiente può essere libero, occupato da un muro, dall’agente o da un oggetto, ma non occupato contemporaneamente da entità diverse. Gli oggetti sono associati a lettere dell’alfabeto, tutte diverse.

Occorre analizzare l’ambiente, suddividendolo in parti. Definiremo *porta* un punto non occupato da muri, ma adiacente a due posizioni occupate da muri sulla stessa riga o sulla stessa colonna (a questo proposito, i punti occupati dall’agente o da oggetti vengono considerati vuoti). In particolare, daremo per scontato che le mappe abbiano sempre una e una sola porta, detta *uscita*, sui bordi dell’ambiente, cioè sulla prima o ultima riga o colonna. Definiremo invece *camera* una regione connessa massimale non occupata da muri o porte, ancora una volta ignorando l’agente e gli oggetti. Il concetto di connessione deriva naturalmente da quello di adiacenza fra punti. Individuate le camere, bisogna determinare per ciascuna le seguenti informazioni. L’area da una camera è il numero di punti che le appartengono. Il perimetro di una camera è il numero di posizioni adiacenti alla camera e occupate da muri o porte. Il punto di riferimento di una camera è quello che le appartiene e ha indice di riga minimo; in caso ve ne sia più d’uno, è quello con indice di colonna minimo.

Conclusa l’analisi dell’ambiente, occorre determinare la gestione dell’agente. Questo può muoversi un passo per volta dal punto corrente a uno adiacente che non sia occupato da muri. Quindi, ha al massimo quattro direzioni possibili, indicate convenzionalmente come *nord* (colonna uguale e riga precedente), *est* (riga uguale e colonna seguente), *sud* (colonna uguale e riga seguente) e *ovest* (riga uguale e colonna precedente). Quando l’agente si trova in un punto adiacente a un oggetto, può “spingerlo”, cioè prenderne il posto e spostarlo di un passo nella stessa direzione, a patto che la nuova posizione dell’oggetto sia libera. Da queste limitazioni derivano diverse conseguenze: i) l’agente e gli oggetti non possono penetrare nei muri; ii) un oggetto adiacente a un muro può strisciare lungo il muro, ma non allontanarsene, perché non esiste una posizione libera fra oggetto e muro in cui infilarsi per spingere; iii) un oggetto in un angolo è bloccato; iv) l’agente non può spostare un treno di due o più oggetti. L’agente deve spostare gli oggetti uno per uno, in

ordine alfabetico crescente, fino all'uscita, senza interrompere le operazioni su uno per agire su un altro. Questo introduce un'ulteriore limitazione: dato che non si possono spostare, gli oggetti successivi si comportano come muri. Gli oggetti precedenti, ovviamente, non sono più nell'ambiente. Al termine del trasloco, l'agente torna automaticamente nella posizione iniziale. Poiché si può svolgere questo compito in molti modi diversi, bisogna determinare il meno costoso. Se l'agente è libero, uno spostamento di un passo ha un costo pari a 1 in direzione nord, 2 in direzione sud, 3 in direzione est e 4 in direzione ovest. Se invece l'agente sta spingendo un oggetto, uno spostamento di un passo dell'agente e dell'oggetto ha un costo pari a 8 in direzione nord, 10 in direzione sud, 13 in direzione est e 15 in direzione ovest.

È intuitivo che il problema riguardi la ricerca di un cammino di costo minimo in un grafo. Tuttavia, la presenza degli oggetti non consente di usare il semplice grafo che associa nodi ai punti dell'ambiente e archi alle coppie di nodi adiacenti. Infatti, via via che gli oggetti si spostano, lo stesso nodo potrebbe essere raggiungibile o no, e lo stesso arco utilizzabile o no. Inoltre, il costo di un arco cambia a seconda che venga percorso dall'agente libero o spingendo un oggetto. Si costruisca un grafo astratto che associ un nodo a una coppia di punti occupati dall'agente e dall'oggetto corrente, e un arco ad ogni spostamento fra nodi che rispetti le regole. È un grafo molto ridondante e ne deriva un algoritmo non molto efficiente, ma ai fini dell'esame va bene.

Il progetto

Il progetto prevede di realizzare un programma che legga da un file di testo la mappa dell'ambiente. Questa ha la forma di una matrice rettangolare di caratteri che corrispondono ai punti della griglia: uno spazio bianco indica che la posizione è libera, una X che è occupata da un muro, una A dall'agente, una lettera minuscola da un oggetto¹. Il seguente esempio riporta una mappa estesa su 8 righe e 11 colonne. Se gli indici di riga e colonna partono da 0², l'uscita è in posizione (0, 1), l'agente in posizione (5, 6) e tre oggetti, etichettati come f, h e m, sono rispettivamente in posizione (2, 5), (6, 3) e (5, 8).

```
X XXXXXXXXX
X XX      X
X XXXf XXXX
X          X
X XXXXXXXX X
X      A m X
X  h      X
XXXXXXXXXXXX
```

Analisi del labirinto Il primo compito richiede di individuare le camere del labirinto, distinguendo fra muri, porte e posizioni interne alle camere, comprese quelle occupate dall'agente e dagli oggetti. L'esempio mostra alcuni casi "degeneri" per chiarire le definizioni. Le posizioni (1, 1), (1, 4), (1, 7) e molte altre sono per definizione porte, anche se parrebbe più naturale definirle "tratti di corridoio", "nicchie" o "porzioni di una camera stretta".

Determinate le camere, bisogna stamparne le informazioni a video. Si indichi il numero delle camere, seguito da uno spazio e dalla parola chiave **camere**. Poi si vada a capo e si riporti in ciascuna riga i seguenti dati di una camera, separati da singoli spazi:

¹Si può assumere una lunghezza massima per le righe della matrice per semplificare la lettura, ma sfruttare questa ipotesi nelle analisi di complessità verrà valutato come errore.

²Questo permette di trattare le righe come stringhe, volendo: non è obbligatorio.

1. la parola chiave **camera** seguita dal numero d'ordine, crescente da 1 al numero totale delle camere;
2. la parola chiave **area** seguita dall'area della camera;
3. la parola chiave **perimetro** seguita dal perimetro della camera;
4. la parola chiave **posizioni** seguita dall'elenco delle posizioni interne alla camera in ordine di riga crescente e, a pari riga, di colonna crescente (in questo modo, la posizione di riferimento è automaticamente in testa all'elenco); ogni posizione va indicata con la coppia degli indici di riga e colonna, racchiusi fra parentesi tonde e separati da una virgola.

Le camere vanno ordinate per area decrescente. In caso di aree coincidenti, si procederà per perimetro decrescente; in caso di perimetri coincidenti, per punto di riferimento successivo, considerando prima indici di riga crescenti e poi indici di colonna crescenti. Nell'esempio considerato:

```

4 camere
camera 1 area 18 perimetro 22 posizioni (5,1) (5,2) (5,3) (5,4) (5,5) ...
camera 2 area 6 perimetro 10 posizioni (1,5) (1,6) (2,5) (2,6) (3,5) (3,6)
camera 3 area 1 perimetro 4 posizioni (3,1)
camera 4 area 1 perimetro 4 posizioni (3,9)

```

dove la riga associata alla prima camera è stata troncata per mancanza di spazio e le ultime due camere, benché innaturali, derivano dalla definizione adottata.

Il secondo compito richiede di muovere l'agente in modo da raggiungere ciascun oggetto e spingerlo sino all'uscita con il costo minimo. Al termine l'agente torna sempre nella sua posizione iniziale. Gli oggetti vanno spostati in ordine alfabetico, con il processo di costo minimo, ignorando il costo del ritorno alla posizione iniziale. In ogni tragitto, gli oggetti precedenti non compaiono più, mentre i seguenti sono inamovibili.

Si dovrà stampare a video il percorso riportando una riga per ciascun oggetto con le seguenti informazioni, separate da spazi:

- l'etichetta dell'oggetto;
- il costo dello spostamento;
- il tragitto percorso dall'agente, rappresentato come una sequenza continua di lettere:
 - **n** e **N** indicano che l'agente si muove verso nord, rispettivamente in modo libero o spingendo un oggetto;
 - **e** e **E** indicano che l'agente si muove verso est, rispettivamente in modo libero o spingendo un oggetto;
 - **s** e **S** indicano che l'agente si muove verso sud, rispettivamente in modo libero o spingendo un oggetto;
 - **o** e **O** indicano che l'agente si muove verso ovest, rispettivamente in modo libero o spingendo un oggetto.

Per esempio, se l'agente si sposta verso est di due passi, il primo liberamente e il secondo spingendo l'oggetto **m**, si stampa la seguente riga.

```
m 11 eE
```

Nel caso in cui non sia possibile spostare l'oggetto fino all'uscita (perché è chiuso in una stanza non raggiungibile dall'agente o da cui non si può accedere all'uscita,

o perché è in una posizione che termina necessariamente in uno stallo), si stamperà una riga contenente:

- l'etichetta dell'oggetto;
- la parola chiave `bloccato`

La soluzione dell'esempio considerato è:

```
f 191 seeennnooonnoSes0000eeeeeeesssoonooooonNNNN
h bloccato
m 159 seen0000000soNNNNNN
```

4 camere

camera 1 area 18 perimetro 22 posizioni (5,1) (5,2) (5,3) (5,4) (5,5)
(5,6) (5,7) (5,8) (5,9) (6,1) (6,2) (6,3) (6,4) (6,5) (6,6) (6,7) (6,8)
(6,9)

camera 2 area 6 perimetro 10 posizioni (1,5) (1,6) (2,5) (2,6) (3,5) (3,6)

camera 3 area 1 perimetro 4 posizioni (3,1)

camera 4 area 1 perimetro 4 posizioni (3,9)

dove le informazioni sulla prima camera vanno a capo per motivi di spazio, ma dovrebbero invece proseguire di seguito.

Dei tre oggetti, il secondo è bloccato, mentre il primo e il terzo vengono spostati verso l'uscita con due tragitti che partono dalla posizione iniziale dell'agente e terminano all'uscita.

```
f 158 seeennnooonnoSes0000eeeeeeesssoonooooonNNNN
h bloccato
m 164 seen0000000soNNNNNN
```

Si noti come ogni volta che l'oggetto deve cambiare direzione, l'agente si muova liberamente per portarsi nella nuova posizione adatta a spingere l'oggetto, a volte con deviazioni complicate (è il caso dell'ultima svolta dell'oggetto `f`).

Chiarimenti

In questa sezione saranno riportate le risposte a domande e dubbi.

Nel caso in cui per poter portare l'oggetto a fuori della stanza serva spostare l'oggetto `b` (per esempio, se questo si trova sull'unica porta della stanza in cui si trova l'oggetto `a`, è lecito per l'agente muovere l'oggetto `b` o gli oggetti ancora da spostare sono come dei muri?

Gli oggetti vanno mossi in ordine alfabetico, per cui quelli successivi (`b` nell'esempio) sono come dei muri. Gli oggetti precedenti, ovviamente, non sono più nel labirinto.