

Problem Set 2

Algoritmi e Strutture Dati – a.a. 2023/2024

Università di Roma “Tor Vergata”

Prof. Luciano Gualà

Linee guida per la consegna. Gli elaborati dovranno essere consegnati entro domenica 21/01/2024 alle ore 23:59. Si prega di seguire le seguenti indicazioni:

1. le soluzioni dovranno essere consegnate via email a entrambi gli indirizzi `guala@mat.uniroma2.it` e `alessandrostr95@gmail.com`;
2. l’oggetto dell’email dovrà essere “Consegna problem set 2 - 2023/2024”;
3. il file dovrà essere in formato pdf;
4. il nome del file dovrà essere “PS_2_XXX.pdf” dove XXX è il primo cognome (in lower case) in ordine alfabetico dei membri del gruppo;
5. l’email dovrà essere inviata tramite l’indirizzo dello studente XXX presente nel nome del file;
6. inserire nel file nome, cognome, indirizzo email e matricola di tutti i membri del gruppo (meglio ancora se tutti i membri del gruppo sono in indirizzo nell’email della consegna).

Buon lavoro 🍀 😊!

Problema 1 (*un nuovo videogioco*)

In questi primi giorni del 2024 è finalmente uscito un nuovo videogioco, *Simultaneous golf*. Il gioco ha tre livelli di difficoltà.

Livello 0: algoritmisti in erba. In questo livello, il campo da golf è una griglia C di dimensione $n \times m$. Ogni cella della griglia può essere una cella d'acqua o una cella di prato. Inizialmente la pallina è posizionata su una cella s di prato e il vostro obiettivo è mandarla in buca. La buca si trova in una cella t . Ad ogni cella di prato è associato un intero positivo. Se la pallina si trova in una cella di prato di valore x potete colpirla nelle quattro direzioni e la pallina finirà nella cella che si trova esattamente x celle più avanti rispetto alla direzione in cui l'avete colpita. Se la pallina finisce in una cella d'acqua avete perso. Le celle che sono fuori dalla griglia potete considerarle tutte celle d'acqua (e quindi non potete mandare la pallina fuori dal campo). Il vostro obiettivo, ovviamente, è mandarla in buca con il minimo numero di mosse.

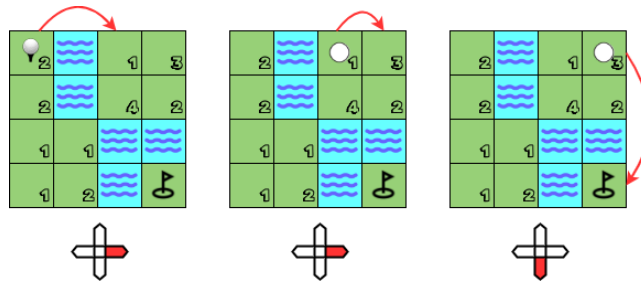


Figure 1

Progettate un algoritmo con complessità $O(nm)$ che calcola una sequenza ottima di mosse o dice correttamente che l'istanza non è risolvibile.

Livello 1: algoritmisti esperti. In questo livello i campi da golf sono *due*, modellati come due griglie C e C' , entrambe di dimensione $n \times m$. Ci sono due palline, posizionate inizialmente in una cella s di C e una cella s' di C' e due buche, t in C e t' in C' . Il vostro obiettivo è giocare *simultaneamente* sui due campi e mandare le palline in buca con il minimo numero di mosse. La dinamica di gioco è la seguente. Se le palline si trovano in due celle di valore x e y rispettivamente, voi potete decidere di colpirle simultaneamente in una delle quattro direzioni: le palline si muoveranno in quella direzione di esattamente x e y celle rispettivamente. Il colpo che date, insomma, è *accoppiato*: ogni volta scegliete una direzione questa sarà la stessa per entrambe le palline. Se una delle due palline (o entrambe) finiscono in una cella d'acqua avete perso.

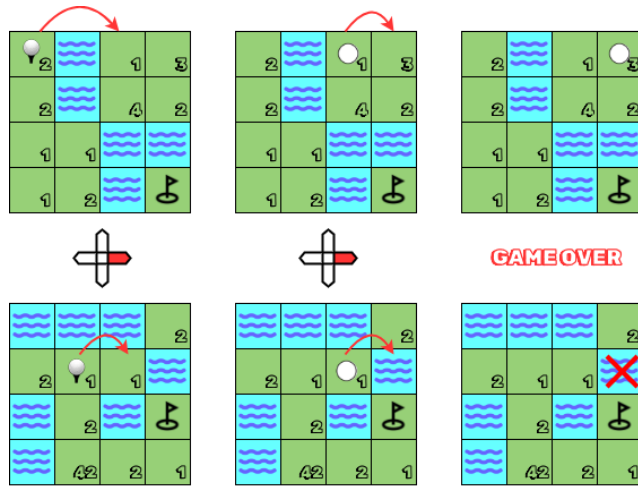


Figure 2: La sequenza di mosse della partita in figura 1 non è più valida per il livello 1.

Vincete se entrambe finiscono in buca (non necessariamente simultaneamente: se una pallina finisce in buca dovete poi mandare l'altra in buca giocando sul campo che vi resta come se foste nel livello precedente).

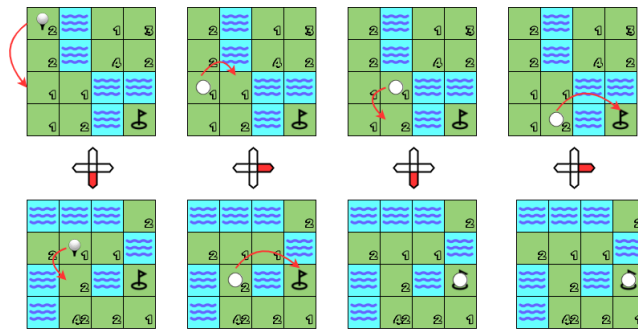


Figure 3: Sequenza vincente per una partita di livello 1.

Progettate un algoritmo efficiente che trova una sequenza ottimale di mosse o dice correttamente che l'istanza non è risolvibile.

Livello ϕ : algoritmisti aurei. Questo livello è come il precedente ma durante la partita avete a disposizione una mossa speciale, che potete usare però solo una volta. La mossa speciale consiste nel *disaccoppiare* un colpo, decidendo

quindi di spostare una sola delle due palline (a vostra scelta). Più precisamente, se le due palline sono rispettivamente in una cella di valore x e una di valore y e decidete di usare la mossa speciale, potete spostare la prima pallina di x celle in una delle quattro direzioni lasciando la seconda pallina ferma, o spostare la seconda pallina di y celle in una delle quattro direzioni lasciando la prima ferma.

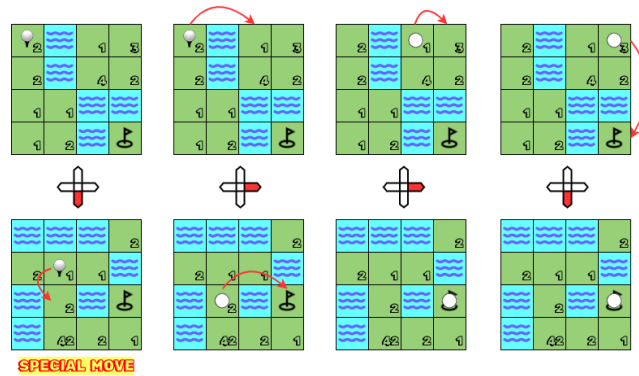


Figure 4: Sequenza vincente per una partita di livello ϕ .

Progettate un algoritmo efficiente che trova una sequenza ottimale di mosse o dice correttamente che l'istanza non è risolvibile.

Problema 2 (*Il dot. Straziota va a cena da amici*)

Il dot. Straziota non si presenta mai a cena da un amico senza aver prima comprato una bottiglia di vino. Ci sono tante enoteche in città e lui ha tanti amici e chissà chi lo inviterà a cena oggi. Questo esercizio vi chiede di aiutarlo a pianificare la miglior strada che dovrà fare da casa sua alla casa di amici per arrivare a cena con una bottiglia di vino.

Più formalmente, sia $G = (V, E, c)$ un grafo non orientato e pesato di n nodi ed m archi, dove ad ogni arco e è associato un costo $c(e) \geq 0$. Sia inoltre $W \subseteq V$ un insieme di nodi detti *enoteche*. Un *wine path* da un nodo s ad un nodo t è un cammino da s a t in G , non necessariamente semplice¹, che attraversa almeno un'enoteca, ovvero un nodo di W . Il costo di un wine path da s a t è definito come la somma dei costi degli archi che attraversa (un arco attraversato più volte viene sommato più volte). Il costo di un wine path di costo minimo è detto *wine distance*.

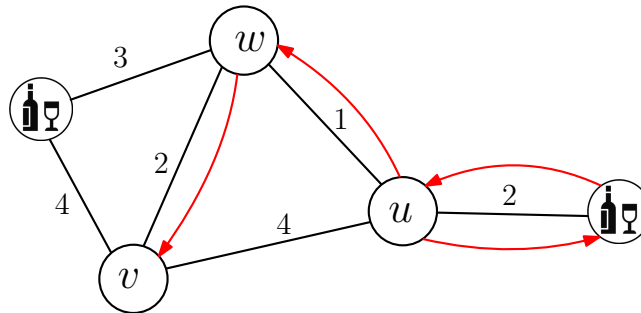


Figure 5: Un wine path da u a v di costo $2 + 2 + 1 + 2 = 7$, che è anche la wine distance da u a v .

- Progettate un algoritmo che preso G , W e un nodo sorgente s , calcola le wine distance da s ad ogni altro nodo v di G . L'algoritmo deve avere complessità temporale $O(m + n \log n)$.
- Dimostrate che esiste un sottografo H di G con $O(n)$ archi che contiene un wine path minimo da s a ogni altro nodo di G .

¹Si ricorda che un cammino è detto *semplice* se non attraversa alcuno nodo più di una volta.