

Algoritmi greedy a Barbieland (**barbie**)

Testo del problema

Slides originali su: judge.science.unitn.it/slides/asd23/prog1.pdf

Barbieland

A Barbieland tutto scorre in maniera tranquilla ed efficiente. I suoi abitanti sono tutti felici, e la vita è semplice e priva di controesempi che invalidano soluzioni basate su tecniche greedy. Infatti, ogni abitante è convinto che prendere di volta in volta la decisione più invitante porti sempre a risolvere correttamente tutti i propri problemi. Tuttavia, un giorno Barbie si sveglia con un dilemma: *e se non fosse vero che gli algoritmi greedy portino sempre a una soluzione ottima?* Questa domanda la porta verso una crisi interiore non facile da gestire.

Rivelazione

Barbie decide di cercare una risposta e si affida a *Barbie Professoressa di Algoritmi*, nota per essere l'abitante di Barbieland più anticonformista in fatto di algoritmi. Barbie Professoressa di Algoritmi le confessa che, in realtà, i problemi risolvibili con tecniche greedy sono solo una piccola parte, e c'è un mare di problemi lì fuori che necessita di soluzioni più sofisticate. Dopo questa rivelazione, Barbie Professoressa di Algoritmi invita Barbie ad esercitarsi, per padroneggiare l'arte del problem solving. Le consiglia quindi di dirigersi verso Algoritmia, un posto pieno di nuovi problemi da scoprire e non limitato dalle tecniche che conosce già.

Algoritmia

Il Mondo di Barbie è infatti composto da una serie di città. Ogni coppia di città può essere collegata o meno da una strada con un certo tempo di percorrenza. Da una città si possono raggiungere sempre tutte le altre, potenzialmente attraversando una o più città intermedie. Qui Barbieland è indicata con B, Algoritmia con A e il resto delle città con un numero identificativo. I tempi di percorrenza sono interi positivi. Dato il viaggio da affrontare, Barbie Professoressa di Algoritmi presta a Barbie la sua auto personale: la *GraphCruise*. Questa automobile ha una particolarità: una volta selezionata la città di partenza e la città di destinazione, ci arriverà seguendo sempre un percorso di costo minimo tra tutti i percorsi possibili. Inoltre, regala a Barbie un libro da cui studiare nuove tecniche algoritmiche.

Un nemico

Tutto sembra perfetto. Tuttavia, il famigerato gruppo nemico degli algoritmi efficienti, la *Miseri Algoritmi Tremendamente Tediosi E Lenti*, ha ricevuto una soffiata ed è pronta a sventare il piano di Barbie. La M.A.T.T.E.L. ha inviato alcuni scagnozzi in giro per le varie città. Se la GraphCruise passasse da una città occupata, verrebbe fermata immediatamente. Ovviamente, la GraphCruise ha le sue regole, che seguirà alla perfezione: la macchina sceglierà uno tra i percorsi più brevi considerando tutta la mappa. Potrebbe decidere quindi autonomamente di dover passare su una città occupata, nel caso in cui esistesse un percorso di costo minimo che lo prevede.

Data la guida autonoma della GraphCruise, Barbie passerà il viaggio studiando dal libro ricevuto. Ovviamente, più tempo studierà, più sarà preparata all'arrivo. Fortunatamente, Barbie Professoressa di Algoritmi ha un potere segreto: è in grado di allungare a piacimento tutte le strade del Mondo di Barbie in un colpo solo, ma tutti gli incrementi dei tempi di percorrenza delle strade saranno dello stesso valore K .

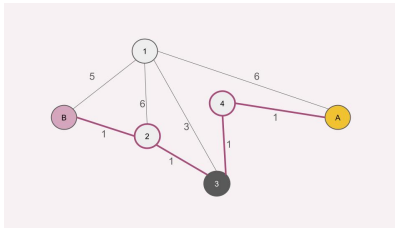


Figura 1: Mappa di esempio: Qui il percorso minimo è segnato in rosa, e passerebbe per una città occupata.

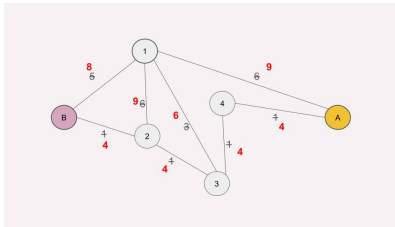


Figura 2: Mappa di esempio: Barbie Professoressa di Algoritmi ha aumentato il tempo di percorrenza di ogni strada di 3.

Domanda

Barbie Professoressa di Algoritmi cercherà di usare il suo potere in aiuto di Barbie, permettendole di passare più tempo possibile studiando in macchina. Aiutatela nel suo lavoro.

- Qual è il massimo valore di K che Barbie Professoressa di Algoritmi può scegliere in modo che il percorso scelto dalla GraphCruise debba passare necessariamente per città non occupate?
- Inoltre, qual è un percorso di costo minimo quando la Professoressa decide di non usare il suo potere?

Esempi

Esempio 1: tutte strade con costo 1

Nell'esempio riportato in Figura 3, ci sono 11 città e 15 strade, le strade hanno tutte tempo di percorrenza 1, i nodi grigi sono quelli occupati, Barbieland è il nodo rosa, Algoritmia è il nodo giallo. Il percorso più breve (costo 2) è quello segnato in rosa, è unico e non passa per nodi grigi. Qualsiasi valore di K non cambierebbe il percorso minimo: K è illimitato (output -1).

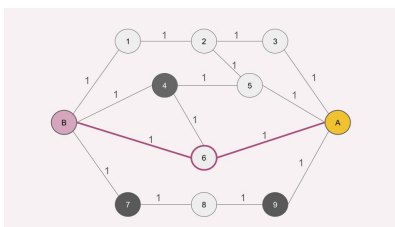


Figura 3: Caso con tutti pesi uguali: il percorso minimo è quello rosa e K è illimitato.

Esempio 2: K non esistente

Figura 4, Grafo uguale a prima, ma il nodo 6 ora è occupato. Non ci sono altri percorsi minimi rispetto a $B-6-A$. Qualsiasi valore di K non cambierebbe il fatto che $B-6-A$ sia il percorso minimo: output -2 .

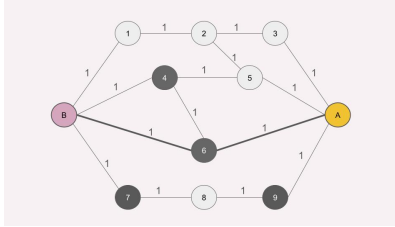


Figura 4: Caso in cui K non esiste: il percorso minimo passa sempre per nodi grigi.

Esempio 3: Tutte città libere

Nel caso in Figura 5 non ci sono città occupate. Ovviamente qualsiasi valore di K va bene, quindi K è illimitato (output -1). Il percorso minimo quando $K = 0$ è $B-4-5-A$ di costo 6.

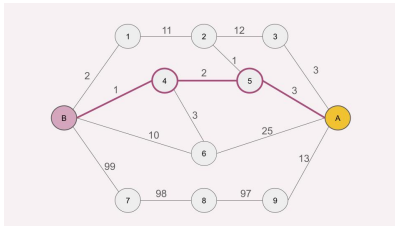
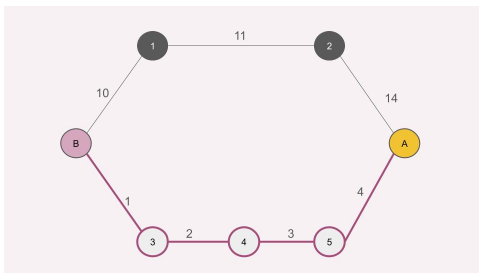


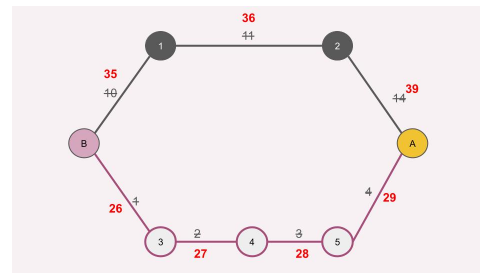
Figura 5: Caso in cui non ci sono città occupate: K è illimitato.

Esempio 4: Percorsi disgiunti

Nel caso in Figura 6a il percorso minimo ha costo 10 e prevede di passare solo per nodi liberi. Esiste un altro percorso per andare da B ad A che ha costo 35, e prevede di passare per nodi occupati. Selezioniamo $K = 25$ e aggiorniamo tutti i pesi (rosso in Figura 6b). A questo punto i due percorsi sono entrambi di costo minimo 110. Quindi Barbie rischierebbe di prendere un percorso minimo che passa per una città occupata. $K = 24$ risulta il massimo valore di K per cui è garantito che Barbie passi solo per nodi non occupati.



(a) Un percorso minimo in rosa.



(b) Tempi di percorrenza aumentati con $K = 25$.

Figura 6: Esempio con percorsi disgiunti.

Esempio 5: Caso generale

In Figura 7 proponiamo un esempio più generale. Con $K = 0$ abbiamo che il percorso minimo è unico e passa per città non occupate. Il percorso è $B-2-3-4-A$ ed ha costo 4. Esistono comunque altri percorsi per andare da B ad A , ma non di peso minimo ed eventualmente passando per città occupate.

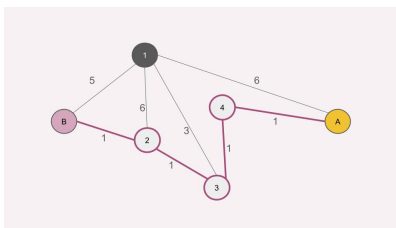
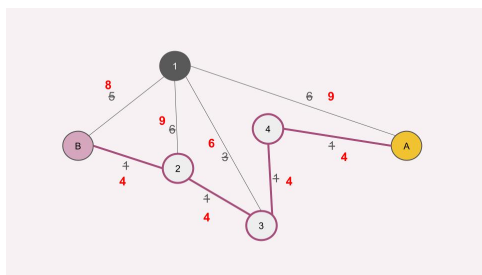
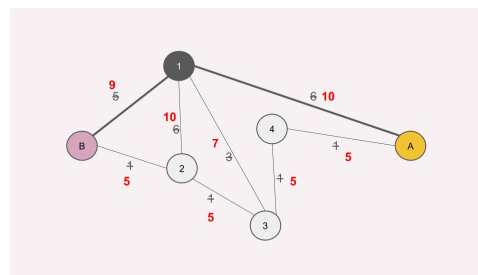


Figura 7: Caso generale: il percorso minimo è quello in rosa.

Proviamo a scegliere $K = 3$ ed aggiorniamo i pesi (rosso, Figura 8a). Il percorso minimo rimane $B-2-3-4-A$, ma ora ha costo 16. Proviamo a scegliere $K = 4$ ed aggiorniamo i pesi (rosso, Figura 8b). A questo punto il percorso minimo è $B-1-A$ con costo 19. Tuttavia bisogna notare che questo percorso attraversa una città occupata. Non è quindi un valore di K valido. Il percorso $B-2-3-4-A$ ora ha costo 20.



(a) $K = 3$.



(b) $K = 4$.

Figura 8: Esempio generale, due tentativi per K .

Altri esempi di soluzioni errate

In Figura 9a abbiamo un esempio in cui K è illimitato. Bisogna trovare il percorso minimo quando $K = 0$. Supponiamo che abbiate dato in output il percorso $B-1-A$, evidenziato in rosa, di costo 11. Esiste però il percorso evidenziato in nero, che costa 4. Il vostro output è errato, in quanto percorso non minimo.

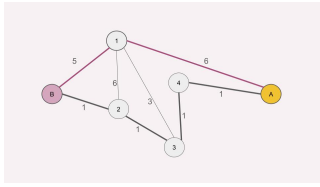
Guardando la Figura 9b, supponiamo abbiate dato come percorso minimo $B-2-A$ e K illimitato. Il percorso è corretto, va bene uno tra $B-2-A$ e $B-1-A$, tuttavia trovare K è impossibile. Per qualsiasi valore di K , entrambi i percorsi manterrebbero lo stesso costo, ergo esisterebbe sempre un percorso di costo minimo con una città occupata.

Figura 9c, esempio che abbiamo già visto. Supponiamo abbiate identificato $K = 2$; con questo valore di K il percorso minimo non passa per alcuna città occupata. Tuttavia K non è massimale, infatti anche per $K = 3$ il percorso di costo minimo non passa per città occupate.

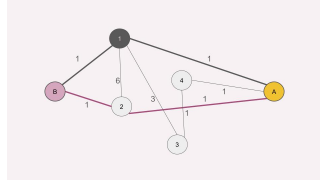
Input/Output

Input: Un file con $1 + S + 1 + M$ righe.

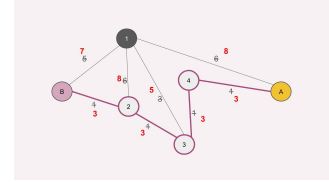
- La prima riga riporta 2 numeri interi: C (int) e S (int), rispettivamente il numero di città del Mondo di Barbie e il numero di strade.



(a) Errore I: percorso sbagliato.



(b) Errore II: percorsi di costo uguale occupati e non.



(c) Errore III: K non è il massimo possibile.

Figura 9: Esempi di soluzioni errate.

- Le successive S righe descrivono le strade. Ogni riga riporta 3 interi a (`int`), b (`int`) e w (`int`), rappresentanti gli identificativi delle due città collegate e il tempo di percorrenza della strada.
- La riga successiva riporta un intero M (`int`): il numero di nodi occupati.
- Le successive M righe riportano ciascuna un intero o (`int`): l'identificativo di una città occupata.

Output: le informazioni da stampare in output constano di una parte obbligatoria e di una facoltativa.

Parte obbligatoria. Questi elementi devono sempre essere presenti nell'output:

1. La prima riga del file di output deve riportare il valore di K (`int`) individuato:
 - Se $K \in \mathbb{N}^+$, riportare K .
 - Se K è illimitato, riportare -1 .
 - Se è impossibile individuare un valore $K > 0$, riportare -2 .

Parte facoltativa. Opzionalmente, l'output può contenere anche le seguenti due righe:

2. La seconda riga contiene un intero R : il numero di città che Barbie visiterebbe se $K = 0$ per raggiungere Algotmia partendo da Barbieland (comprese B e A) seguendo uno qualsiasi tra tutti i percorsi minimi (non importa se si attraversano città occupate);
3. La terza riga contiene R interi: le città che sarebbero visitate secondo questo percorso, in ordine, da Barbie (comprese B e A), se $K = 0$.

Assunzioni e casi di test

Assunzioni

- $2 \leq C \leq 1.000$
- Barbieland è al nodo 0, Algotmia è al nodo $C - 1$
- Ogni altra città è identificata da un numero tra 1 e $C - 2$
- $0 \leq M \leq C - 2$
- Barbieland e Algotmia sono sempre NON occupate
- $0 \leq S \leq 10.000$
- Per il tempo di percorrenza w_i dell' i -esima strada vale $1 \leq w_i \leq 100.000$
- Ogni grafo è indiretto.
- Ogni grafo è connesso.
- Non ci sono self-loop, cioè archi (a, a) dove a è una città
- Non ci sono archi doppi, ovvero esiste al massimo una tripla (a, b, w) per una data coppia non ordinata $\{a, b\}$

Casi di test Ci sono 20 casi di test in totale:

- In 6 casi su 20, tutti gli archi hanno lo stesso tempo di percorrenza;
- In 4 casi su 20, o nessuna città è occupata, o tutte le città sono occupate (B ed A escluse);
- In 4 casi su 20, i percorsi da B ad A sono tutti disgiunti, ovvero non condividono città intermedie;
- Nel restante dei casi non ci sono particolari limitazioni.

I limiti di tempo e memoria sono:

- Tempo limite massimo: 1 secondo;
- Memoria massima: 16 MB.

Punteggi e correttore

Ogni caso di test vale 5 punti se completo (parte obbligatoria e facoltativa), 3 punti se parziale (solo parte obbligatoria). Il punteggio massimo è di 100 punti.

In caso di output errato si prendono sempre 0 **punti**: * se K non è corretto, si ottengono 0 **punti**. * se K è corretto, ma il percorso stampato non è possibile, si ottengono 0 **punti**. * se K è corretto, ma il percorso stampato non è minimo, si ottengono 0 **punti**.

Valutazione

Per valutazione del progetto:

- Conta il punteggio dell'**ultimo sorgente** inviato al sistema;
- Il progetto è superato con un punteggio non inferiore a 30 punti;
- C'è un limite di 50 sottoposizioni per gruppo;

Dataset di esempio

Potete trovare un dataset equivalente con cui esercitarvi in locale:

judge.science.unitn.it/slides/asd23/dataset_barbie.zip.

Nota: Il dataset di esempio contiene in output solo la prima riga, che indica il valore di K , senza ulteriori informazioni.

Esempi di input/output

File input.txt	File output.txt
<pre>6 8 0 1 5 0 2 1 1 2 6 1 3 3 1 5 6 2 3 1 3 4 1 4 5 1 0</pre>	<pre>-1 5 0 2 3 4 5</pre>
<pre>20 31 9 7 7 17 1 6 4 17 3 5 2 9 1 6 6 8 13 9 7 11 2 14 17 3 8 5 10 9 6 7 19 1 7 3 8 1 3 5 3 1 4 8 11 15 1 18 10 6 14 19 8 1 14 8 3 2 5 19 17 7 2 8 1 15 10 3 13 2 9 12 18 6 15 8 9 13 3 9 0 16 9 5 13 3 6 16 6 19 4 9 4 14 2 5 1 7 8 9 12</pre>	<pre>-2 5 0 16 6 1 19</pre>

File input.txt	File output.txt
<pre>7 7 0 1 10 0 3 1 1 2 11 2 6 14 3 4 2 4 5 3 5 6 4 2 1 2</pre>	<pre>24 5 0 3 4 5 6</pre>

File input.txt	File output.txt
<pre>7 10 1 0 3 2 4 3 2 1 3 5 0 1 4 6 6 2 6 8 5 1 1 4 0 8 3 4 5 5 3 9 2 3 4</pre>	<pre>-2 5 0 5 1 2 6</pre>