Algoritmi e Strutture Dati

GTA VI – Povo City (`gta`)

Testo del problema

Slides originali su: asdlab.disi.unitn.it/asd26/prog1.pdf
A. sta giocando a GTA VI quando, osservando il mondo virtuale, commenta: «Questi giochi hanno raggiunto un grado di realismo incredibile, certo però che mi sembra tutto un po' inefficiente…». All'improvviso viene catapultato nel videogioco e si ritrova a Povo City nei panni di *Big O*, noto spacciatore di algoritmi efficienti.
La sua missione è distribuire il suo manuale di algoritmi agli abitanti di Povo City. Per farlo, Big O dispone di alcuni furgoni, ciascuno dei quali parte dal deposito assegnato con una copia del manuale da consegnare a domicilio.
Povo City è costituita da incroci collegati da strade percorribili inizialmente in **entrambi i sensi**. Da ogni incrocio è possibile raggiungere qualunque altro incrocio e tra due incroci può esistere al massimo una strada diretta.
La Polizia Inefficiente si mette sulle tracce di Big O e dei suoi furgoni: vuole rallentare la diffusione degli algoritmi efficienti. Arrestarlo sarebbe troppo efficiente! Per ostacolare la sua missione, la polizia decide quindi di istituire dei sensi unici a richiesta.
Al primo passaggio di un furgone su una strada, una pattuglia mette un posto di blocco all'ingresso opposto. Da quel momento, tutti i furgoni possono percorrere quella strada **solamente nel verso del primo passaggio**. Il senso unico vale fino alla fine della missione; le strade non ancora percorse rimangono a doppio senso.
Prima che vengano istituiti i posti di blocco, alcune strade costituiscono l'unico collegamento tra due zone della città. Big O chiama queste strade **varchi**.

> **Definizione di varco**
>
> Un varco è una strada $u \leftrightarrow v$ che, nella mappa originale di Povo City, rappresenta l'unico modo per raggiungere v partendo da u. Eliminando questa strada, diventa impossibile passare da una delle due zone della città all'altra.

Il vostro compito

Aiutate Big O a portare a termine la sua missione e a diffondere gli algoritmi efficienti a Povo City! Per farlo dovete rispondere alla seguente domanda:

- **È possibile consegnare tutti i manuali?**
  - Se sì, indicate quali varchi i furgoni sono obbligati ad attraversare e in quale direzione devono essere percorsi.
  - Se no, indicate un varco che due consegne richiedono di percorrere in versi opposti.

Vi verrà fornita la mappa di Povo City, contenente N incroci collegati da M strade inizialmente bidirezionali, ciascuna specificata da una coppia (U, V). Vi sarà inoltre indicato il numero P di manuali da consegnare. Per ciascun manuale sarà specificata una coppia (L_p, L_d), che indica rispettivamente l'incrocio (il deposito) di partenza del furgone e l'incrocio di destinazione in cui consegnare il manuale.

Esempi

Esempio 1

Nell'esempio di Figura 1 mostriamo una mappa con 9 incroci, collegati da 10 strade. Ci sono 3 manuali da consegnare: $2 \to 6$, $1 \to 7$ e $8 \to 6$. In questa mappa le strade $0 \leftrightarrow 4$, $4 \leftrightarrow 6$ e $7 \leftrightarrow 8$ sono varchi. Al contrario, $0 \leftrightarrow 1$ non è un varco, perché si può raggiungere 1 partendo da 0 anche tramite il percorso $0 \to 3 \to 2 \to 1$ nella mappa originale.

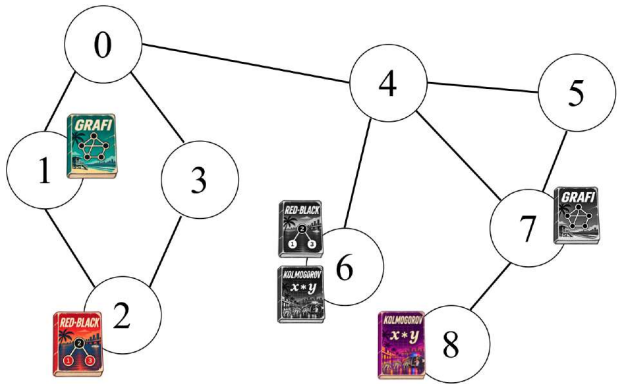
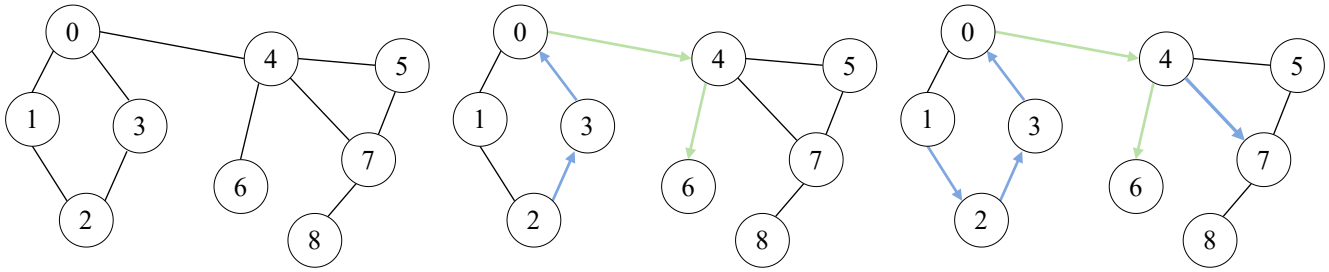


Figura 1: Una mappa di Povo City con 9 incroci, collegati da 10 strade, e 3 manuali da consegnare: $2 \to 6$, $1 \to 7$ e $8 \to 6$.

La Figura 2 riporta una possibile soluzione:

- (a) Il primo furgone consegna il manuale dall'incrocio 2 al 6 seguendo il percorso $2 \to 3 \to 0 \to 4 \to 6$.
- (b) Il secondo furgone consegna il manuale dall'incrocio 1 al 7 seguendo il percorso $1 \to 2 \to 3 \to 0 \to 4 \to 7$.
- (c) Il terzo furgone consegna il manuale dall'incrocio 8 al 6 seguendo il percorso $8 \to 7 \to 5 \to 4 \to 6$.

In questa soluzione, i varchi che i furgoni sono obbligati ad attraversare sono **3**: $\mathbf{0 \to 4}$, $\mathbf{4 \to 6}$ e $\mathbf{8 \to 7}$.



(a) $2 \to 6$ (b) $1 \to 7$ (c) $8 \to 6$

Figura 2: Una soluzione del primo esempio. Le frecce verdi indicano i varchi attraversati, quelle azzurre le altre strade percorse.

Un possibile output è il seguente:

```
1
3
0 4
4 6
8 7
```

Esempio 2

Nell'esempio di Figura 3 mostriamo una mappa con 7 incroci, collegati da 6 strade che formano un albero. Ci sono 2 manuali da consegnare: $4 \to 2$ e $4 \to 3$. In questo caso tutte le strade della mappa sono varchi.
La Figura 4 riporta una soluzione: (a) il primo furgone segue il percorso $4 \to 1 \to 0 \to 2$; (b) il secondo furgone segue il percorso $4 \to 1 \to 0 \to 3$. I varchi che i furgoni sono obbligati ad attraversare sono **4**: $\mathbf{4 \to 1}$, $\mathbf{1 \to 0}$, $\mathbf{0 \to 2}$ e $\mathbf{0 \to 3}$.

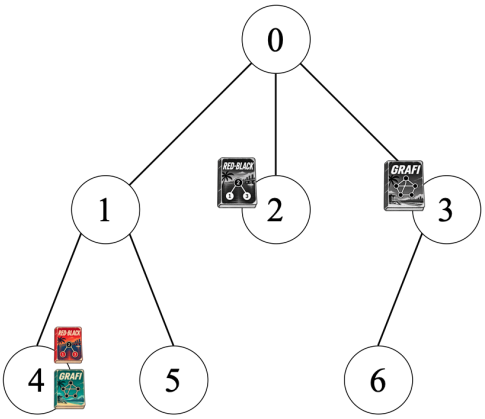
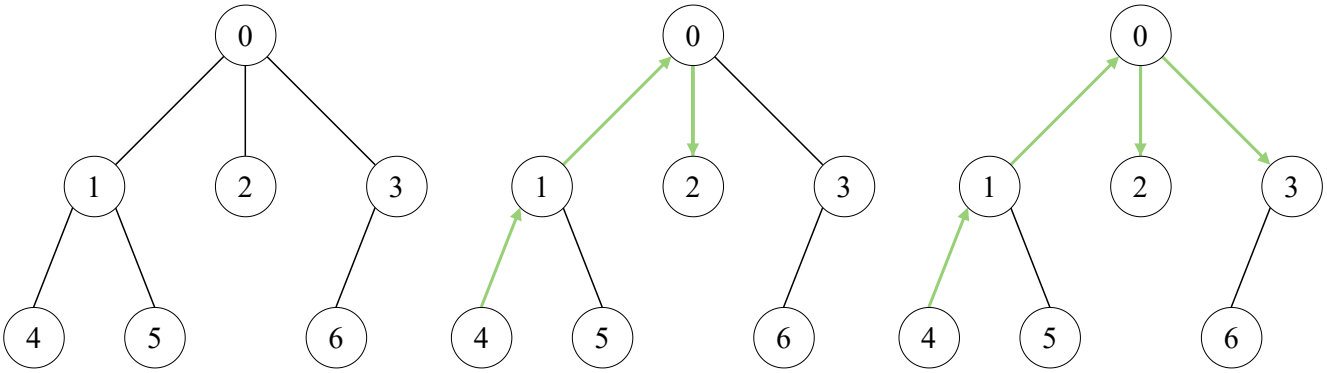


Figura 3: Una mappa di Povo City con 7 incroci, collegati da 6 strade che formano un albero. Ci sono 2 manuali da consegnare: $4 \to 2$ e $4 \to 3$.



(a) $4 \to 2$ (b) $4 \to 3$ (c) $6 \to 0$ (impossibile)

Figura 4: La soluzione del secondo esempio (a, b) e il conflitto che si genererebbe aggiungendo una terza consegna da 6 a 0 (c).

Un possibile output per le due consegne è il seguente:

```
1
4
4 1
1 0
0 2
0 3
```

Se ci fosse un terzo manuale da consegnare da 6 a 0, sarebbe impossibile completare tutte le consegne. Il terzo furgone dovrebbe infatti percorrere il varco $0 \leftrightarrow 3$ nel verso $3 \to 0$, opposto a quello richiesto dal secondo furgone ($0 \to 3$). In questo caso un possibile output sarebbe:

```
0
3 0
```

È ugualmente corretto stampare 0 e, sulla riga successiva, la coppia 0 3.

Esempio 3

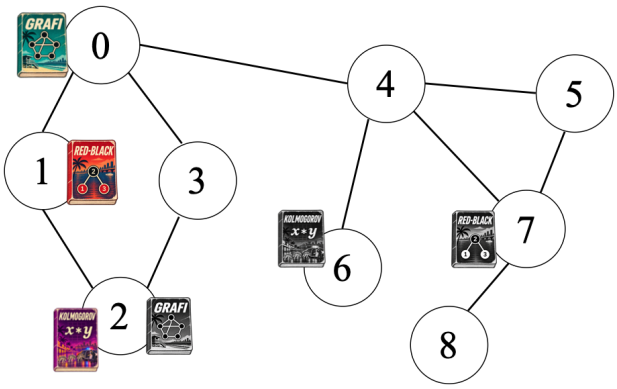
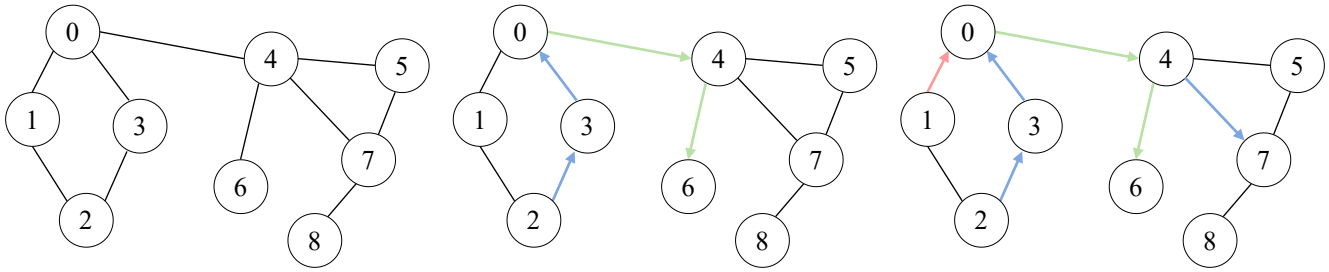


Figura 5: Una mappa di Povo City con 9 incroci, collegati da 10 strade, e 3 manuali da consegnare: $2 \to 6$, $1 \to 7$ e $0 \to 2$.

Nell'esempio di Figura 5 mostriamo la stessa mappa del primo esempio, ma cambiamo le consegne da effettuare: $2 \to 6$, $1 \to 7$ e $0 \to 2$.
La Figura 6 riporta una soluzione *errata*: (a) il primo furgone segue il percorso $2 \to 3 \to 0 \to 4 \to 6$; (b) il secondo furgone segue il percorso $1 \to 0 \to 4 \to 7$. A questo punto il terzo furgone non può più consegnare il manuale dall'incrocio 0 al 2: non può percorrere né la strada $0 \to 1$, né la strada $0 \to 3$, perché entrambe sono diventate a senso unico nel verso opposto.

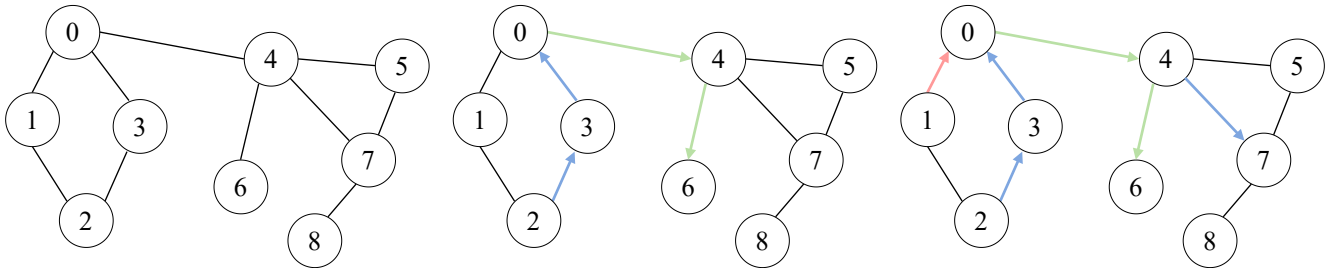


(a) $2 \to 6$ (b) $1 \to 7$ (c) $0 \to 2$ (bloccato)

Figura 6: Una scelta errata dei percorsi del terzo esempio: la seconda consegna impedisce di completare la terza.

Questo non significa, però, che sia impossibile completare tutte le consegne. Esiste una soluzione corretta, mostrata in Figura 7: (a) il primo furgone segue ancora il percorso $2 \to 3 \to 0 \to 4 \to 6$; (b) il secondo furgone segue il percorso $1 \to 2 \to 3 \to 0 \to 4 \to 7$ anziché passare direttamente da 1 a 0; (c) il terzo furgone può ora percorrere $0 \to 1 \to 2$ e completare l'ultima consegna.
In questa soluzione, i varchi che i furgoni sono obbligati ad attraversare sono **2**: $\mathbf{0} \to \mathbf{4}$ e $\mathbf{4} \to \mathbf{6}$. Un possibile output è il seguente:

```
1
2
0 4
4 6
```



(a) $2 \to 6$ (b) $1 \to 7$ (c) $0 \to 2$

Figura 7: Una soluzione corretta del terzo esempio: scegliendo diversamente il percorso del secondo furgone è possibile completare tutte le consegne.

Input/Output

Input: un file con $1 + M + P$ righe.

- La prima riga riporta 3 numeri interi: N (`int`), M (`int`) e P (`int`), rispettivamente il numero di incroci, strade e manuali.
- Le successive M righe riportano le strade che collegano gli incroci. Ogni riga riporta 2 interi U (`int`) e V (`int`), rappresentanti gli incroci collegati.
- Le successive P righe sono costituite da 2 interi ciascuna: L_p (`int`) e L_d (`int`), che indicano rispettivamente l'incrocio (il deposito) di partenza in cui si trova il furgone con il manuale e l'incrocio di destinazione in cui consegnarlo.

Output: le informazioni da stampare in output dipendono dal fatto che sia possibile o meno consegnare tutti i manuali. La prima riga del file di output deve contenere 0 se è **impossibile** consegnare tutti i manuali, 1 se è **possibile**.

- *Caso consegna impossibile.* Se è impossibile consegnare tutti i manuali (caso 0):
  - La seconda riga deve riportare 2 interi U e V rappresentanti **un varco che sarebbe necessario attraversare in entrambe le direzioni**. È indifferente stampare la coppia (U, V) o (V, U). Se ci sono più varchi in conflitto, è sufficiente stamparne uno qualsiasi.
- *Caso consegna possibile.* Se è possibile consegnare tutti i manuali (caso 1):
  - La seconda riga deve riportare il valore K (`int`), ovvero il numero di varchi che i furgoni sono obbligati ad attraversare per consegnare i manuali.
  - Le successive K righe devono contenere 2 interi ciascuna, U e V, rappresentanti la direzione $U \to V$ in cui deve essere attraversato ciascun varco.

Note:

- Vanno stampati **solamente i varchi** che i furgoni sono obbligati ad attraversare. Le strade che non sono varchi e i varchi non attraversati non devono essere stampati.
- L'ordine con cui stampate i K varchi non è importante.

Assunzioni e casi di test

Assunzioni

- $1 \leq N \leq 200.000$

- $1 \leq M \leq 200.000$
- $1 \leq P \leq 200.000$
- Ogni grafo è connesso.
- Ogni grafo è non diretto.

Casi di test

- 20 casi di test in totale;
- In 6 casi, il grafo in input è un **albero** e c'è un solo manuale da consegnare ($P = 1$);
- In 9 casi, il grafo in input non è un albero;
- In 5 casi, il grafo in input è un albero con $N \geq 40.000$ e $P \geq 50.000$.

Per raggiungere la sufficienza è possibile risolvere i 6 casi di test in cui il grafo è un albero e $P = 1$.
I limiti di tempo e memoria sono:

- Tempo limite massimo: 2 secondi;
- Memoria massima: 64 MB.

Punteggi e correttore

Ogni caso di test vale 5 punti. Il punteggio massimo è di 100 punti.
Per ogni caso di test, le soluzioni vengono valutate nel modo seguente:

1. Le soluzioni corrette ottengono 5 **punti**:
   - **consegna impossibile**: stampa 0 e un varco che genera un conflitto;
   - **consegna possibile**: stampa 1, il valore corretto di K e i K varchi obbligatori, orientati nella direzione corretta.
2. Le soluzioni errate ottengono 0 **punti**. Sono soluzioni errate:
   - i casi in cui la risposta, possibile (1) o impossibile (0), non è quella corretta;
   - i casi in cui la risposta corretta è impossibile, ma si stampa un arco che non genera conflitti;
   - i casi in cui la risposta corretta è possibile, ma K non è corretto;
   - i casi in cui la risposta corretta è possibile, ma la lista degli archi orientati data in output non è corretta (ad esempio, contiene strade che non sono varchi, omette varchi obbligatori o li orienta nel verso sbagliato).

Valutazione

- Il progetto è superato con un punteggio non inferiore a 30 punti;
- C'è un limite di 40 sottoposizioni per gruppo.

Punti bonus per l'esame. L'assegnazione dei punti bonus avviene in maniera competitiva:

- 3 punti ai gruppi nel primo terzile della classifica (primo terzo);
- 2 punti ai gruppi nel secondo terzile della classifica (secondo terzo);
- 1 punto ai gruppi nel terzo terzile della classifica (ultimo terzo).

Per l'assegnazione dei punti bonus vengono considerati solamente i gruppi che raggiungono la sufficienza, ovvero un punteggio maggiore o uguale a 30.
La classifica è disponibile all'indirizzo asdlab.disi.unitn.it/arena/ranking/.

Dataset di esempio

Potete trovare un dataset equivalente con cui esercitarvi in locale sul sito asdlab.disi.unitn.it.
Nota: il dataset di esempio contiene in output solo la prima riga, che indica se è possibile (**1**) o impossibile (**0**) completare tutte le consegne, senza ulteriori informazioni.

Esempi di input/output

File `input.txt`	**File** `output.txt`

`9 10 3`
`0 1`
`0 3`
`1 2`
`2 3`
`0 4`
`4 5`
`4 6`
`4 7`
`5 7`
`7 8`
`2 6`
`1 7`
`0 2`
	`1`
	`2`
	`4 6`
	`0 4`

File `input.txt`	**File** `output.txt`

`7 6 2`
`0 1`
`0 2`
`0 3`
`1 4`
`1 5`
`3 6`
`4 2`
`4 3`
	`1`
	`4`
	`4 1`
	`1 0`
	`0 2`
	`0 3`

File `input.txt`	**File** `output.txt`

`7 6 5`
`2 5`
`0 2`
`2 6`
`6 4`
`3 6`
`1 2`
`3 0`
`4 1`
`1 4`
`4 5`
`4 2`
	`0`
	`4 6`