
gta
VI
Povo City

A. prova GTA VI



«Questi giochi hanno raggiunto un grado di realismo incredibile, certo però che mi sembra tutto un po' inefficiente...»



Missione: consegnare il manuale

A. 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**. Ogni furgone parte dal deposito assegnato con una copia da consegnare a domicilio.



Le strade di Povo City

All'inizio tutte le strade sono a doppio senso. Da ogni incrocio si può raggiungere qualunque altro incrocio. Tra due incroci può esserci al massimo una strada diretta.

La Polizia Inefficiente

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!

Al primo passaggio di un furgone su una strada, una pattuglia mette un posto di blocco all'ingresso opposto.



Senso unico a richiesta

Da quel momento, tutti i furgoni possono percorrere quella strada solo **nel verso del primo passaggio**. Il senso unico vale fino alla fine della missione.

Il vostro compito

I varchi di Povo City

Prima dei posti di blocco, alcune strade sono l'unico collegamento tra due zone della città. Se ne chiudete una, diventa impossibile passare da una zona all'altra. Chiameremo queste strade **varchi**.

Si possono completare tutte le consegne?

- Se sì, indicate quali varchi i furgoni sono obbligati ad attraversare e in quale verso.
- Se no, indicate un varco che due consegne richiedono di percorrere in versi opposti.

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 rispettivamente sono l'incrocio (il deposito) di partenza in cui si trova il furgone con il manuale e l'incrocio di destinazione in cui consegnarlo.

Output

La prima riga del file di output deve essere 0 **se è impossibile** consegnare tutti i manuali, 1 **se è possibile**. Successivamente, il formato di output varia in base alla risposta data nella prima riga.

Se **è impossibile** consegnare tutti i manuali:

- La seconda riga deve riportare 2 interi rappresentanti **un varco che dovrebbe essere attraversato in entrambe le direzioni**. È possibile che ci siano molteplici varchi in conflitto, in tal caso è sufficiente stamparne uno qualsiasi.

  Nota: in questo caso è possibile stampare sia $U\ V$ che $V\ U$, è indifferente.

Output

Se **è possibile** consegnare tutti i manuali:

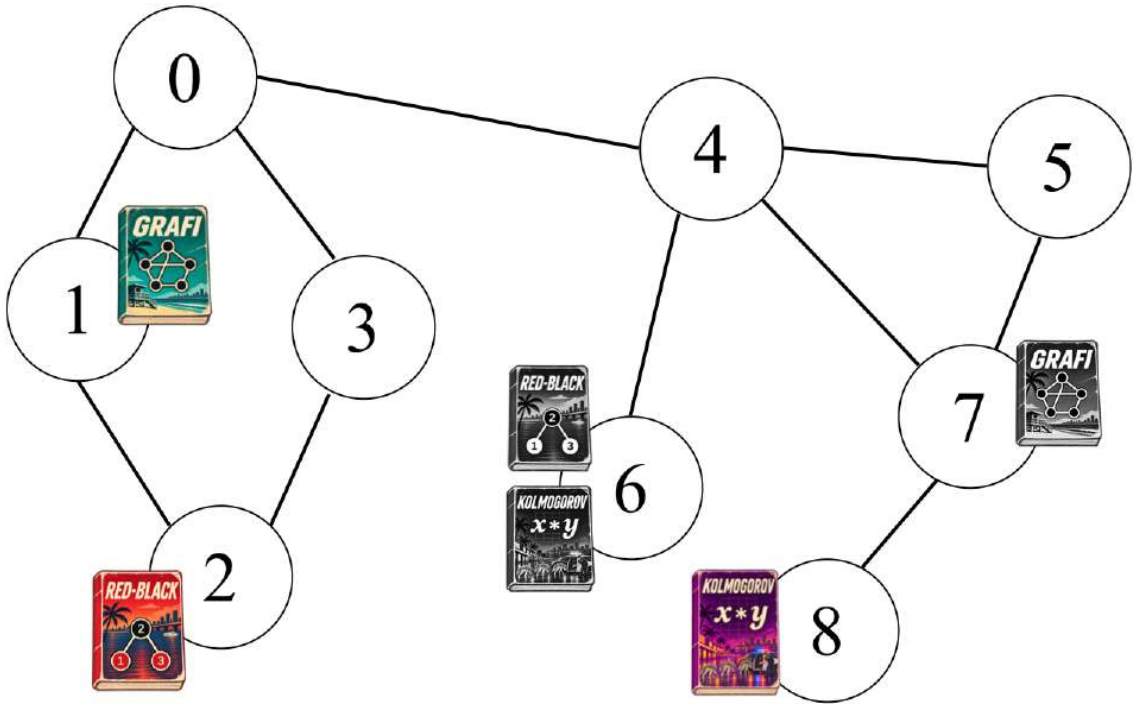
- La seconda riga deve contenere il valore K (`int`), ovvero il numero di **varchi** che i furgoni sono **obbligati ad attraversare** per consegnare i manuali.
- Successivamente, devono essere stampate K righe, ognuna contenente 2 interi U e V, rappresentanti la **direzione** dei varchi attraversati dai furgoni.

  Nota 1: vanno stampati **solamente i varchi** che i furgoni devono attraversare. Le strade che non sono varchi e i varchi non attraversati **non devono essere stampati**.
  Nota 2: l'ordine con cui stampate i K varchi non è importante.

Esempio 1

Manuali da consegnare:

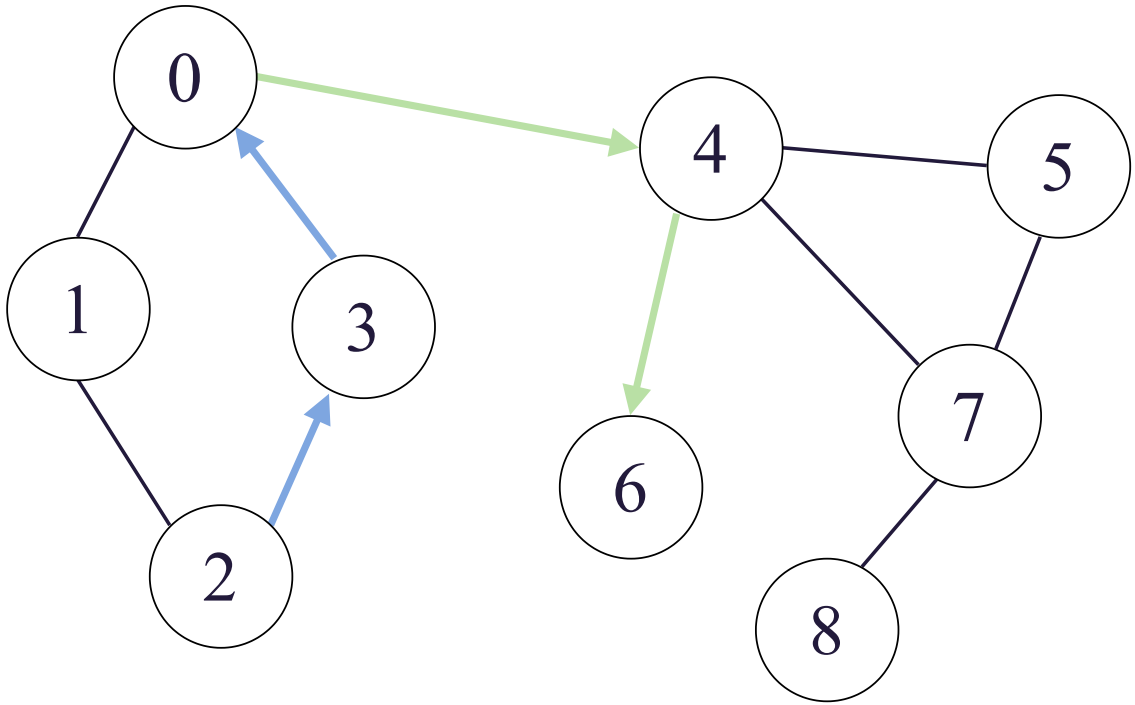
- $2 \to 6$
- $1 \to 7$
- $8 \to 6$



Esempio 1.1

Manuali da consegnare:

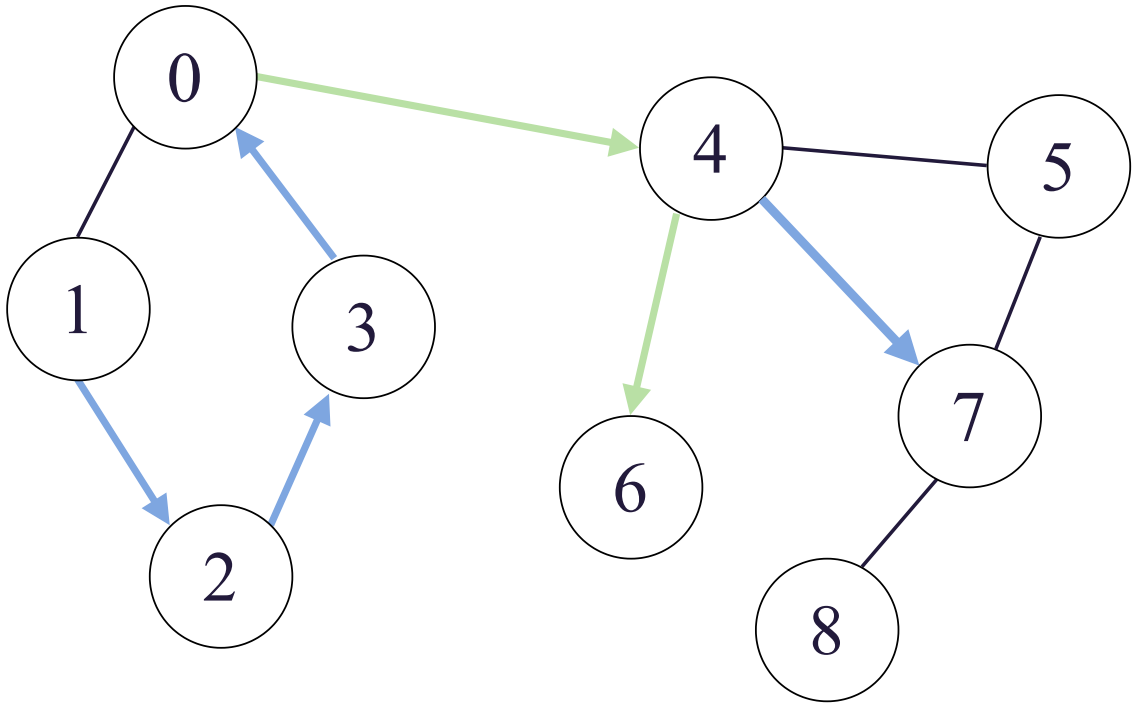
- $2 \to 6$ ✔
- $1 \to 7$
- $8 \to 6$



Esempio 1.2

Manuali da consegnare:

- $2 \to 6$ ✔
- $1 \to 7$ ✔
- $8 \to 6$



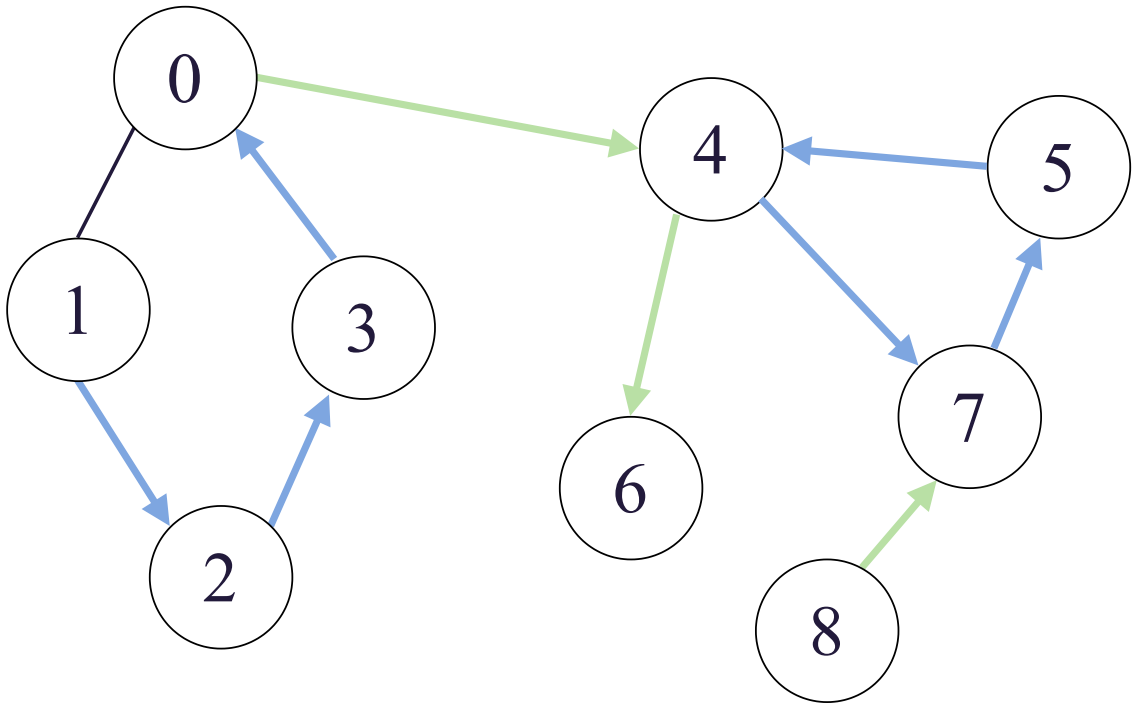
Esempio 1.3

Manuali da consegnare:

- $2 \rightarrow 6$ ✔
- $1 \rightarrow 7$ ✔
- $8 \rightarrow 6$ ✔

In questo caso l'output sarebbe il seguente:

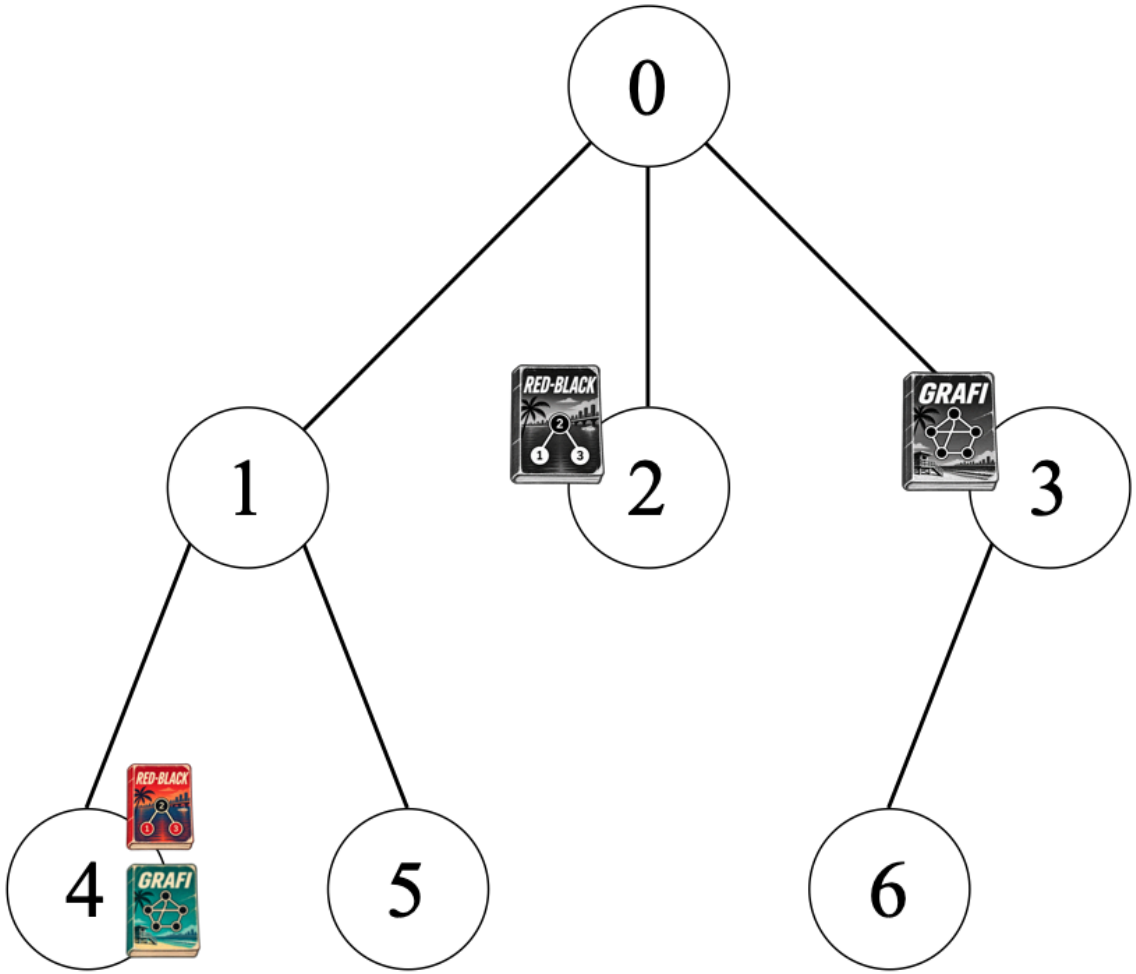
```
1
3
0 4
4 6
8 7
```



Esempio 2

Manuali da consegnare:

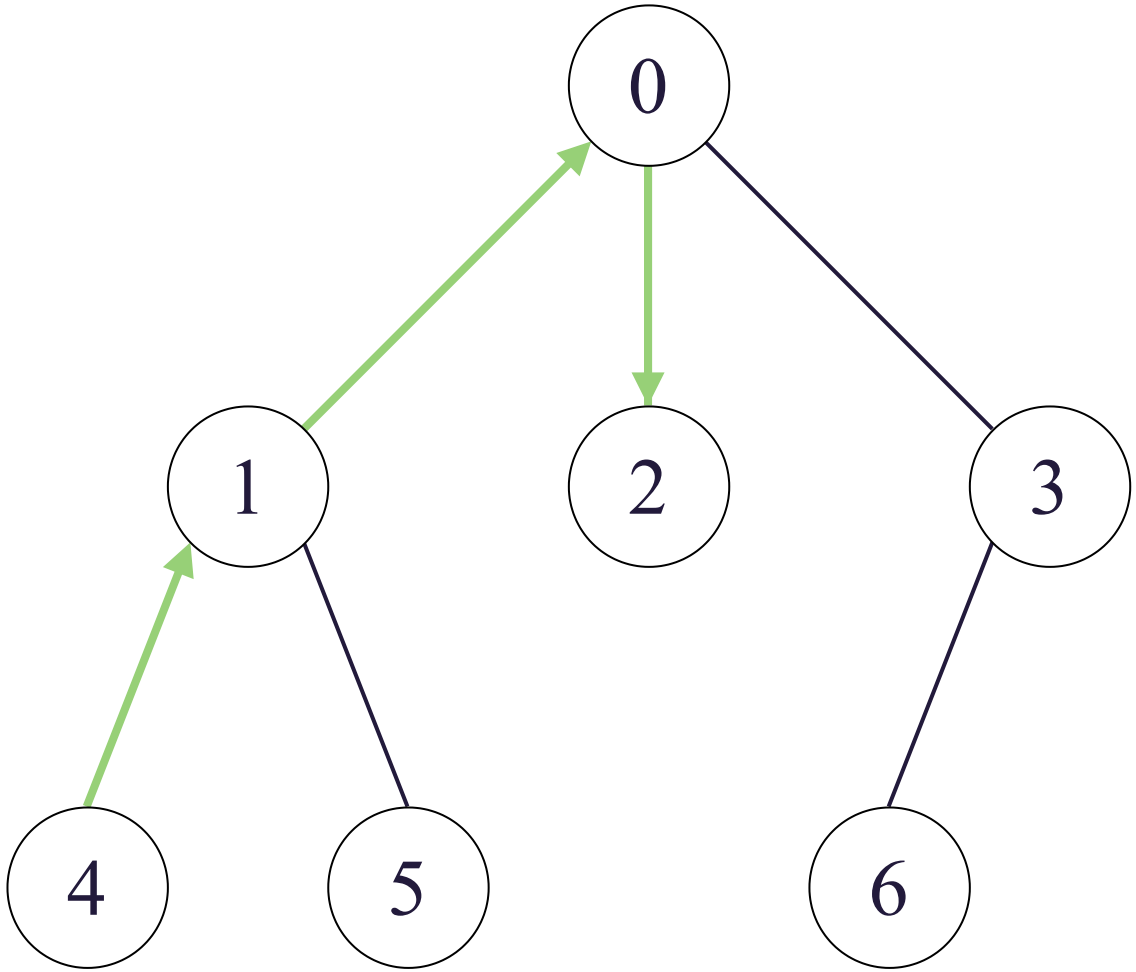
- $4 \rightarrow 2$
- $4 \rightarrow 3$



Esempio 2.1

Manuali da consegnare:

- $4 \to 2$ ✔
- $4 \to 3$



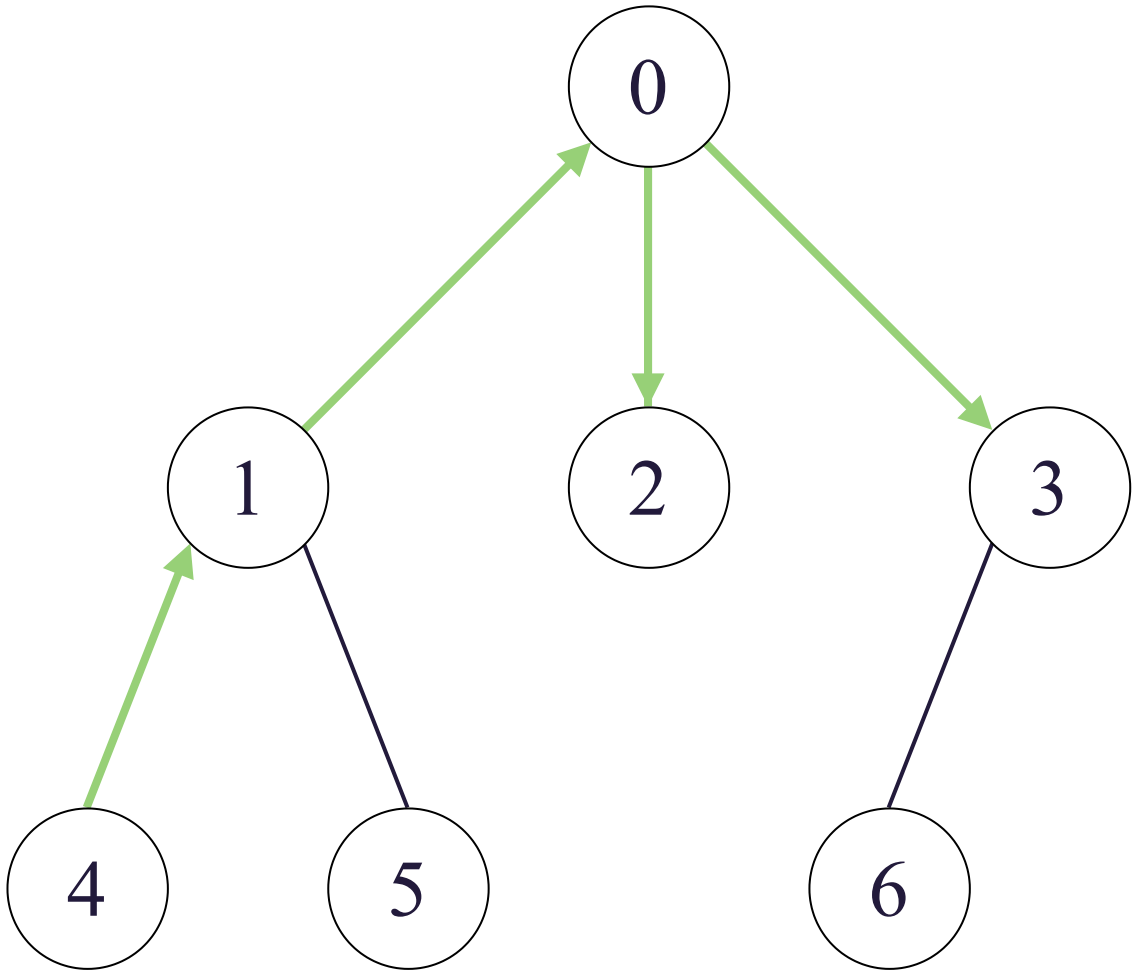
Esempio 2.2

Manuali da consegnare:

- $4 \rightarrow 2$ ✔
- $4 \rightarrow 3$ ✔

In questo caso l'output sarebbe il seguente:

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



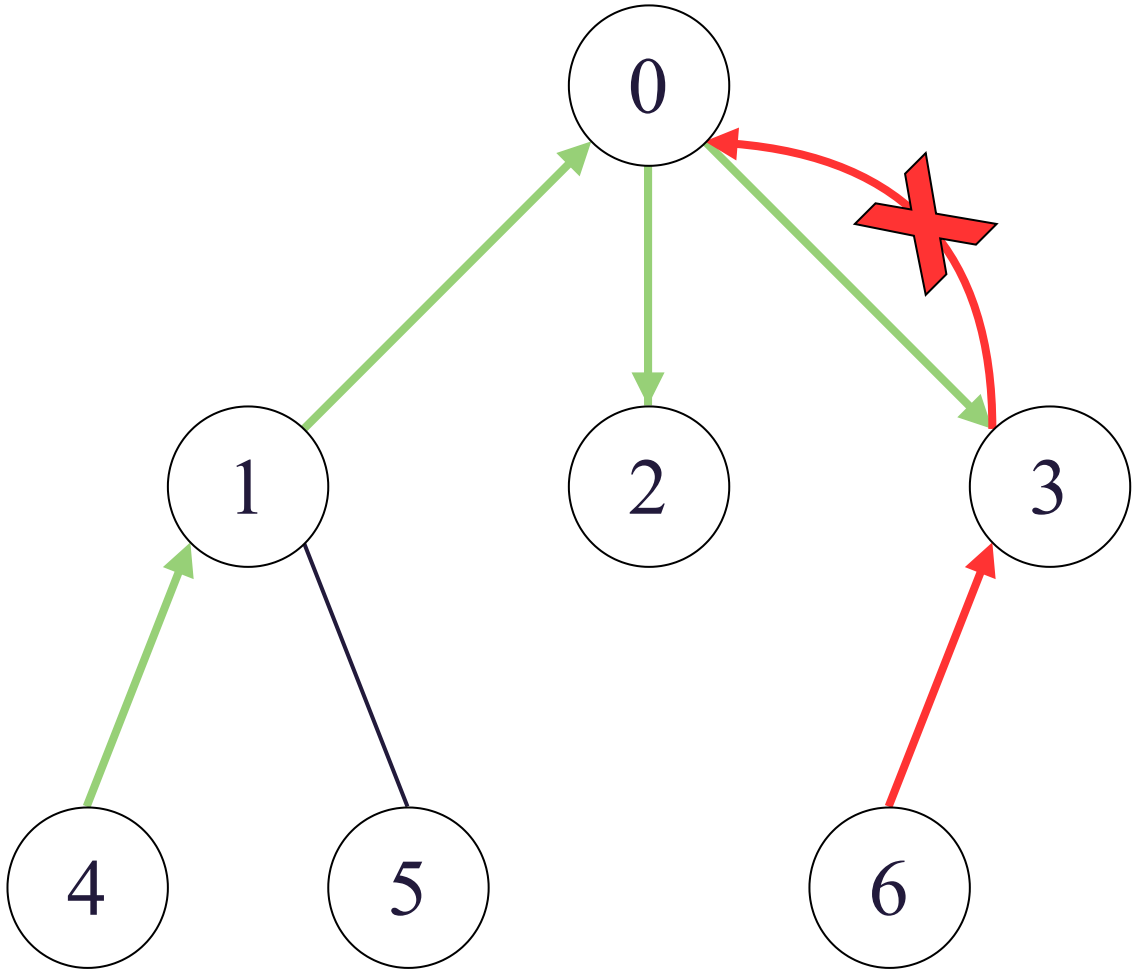
Esempio 2.3

Tuttavia, se ci fosse un terzo Manuale da consegnare:

- $4 \to 2$ ✔
- $4 \to 3$ ✔
- $6 \to 0$ ✗

In questo caso l'output sarebbe il seguente:

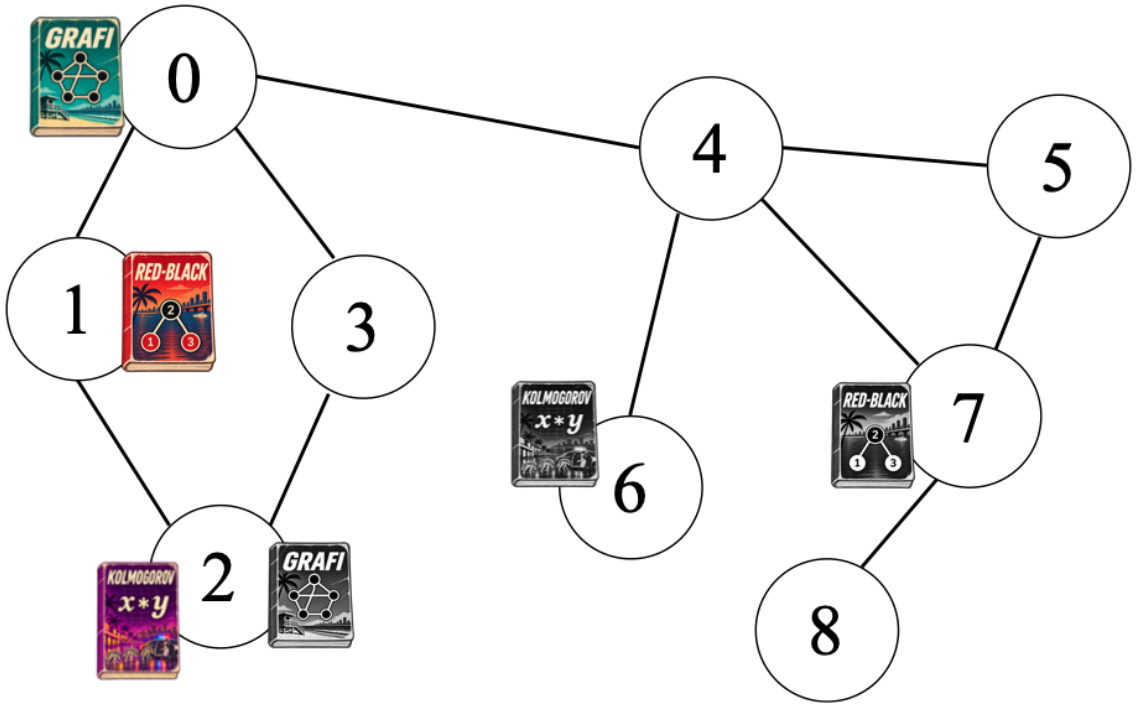
```
0
3 0
```



Esempio 3 - Risposta errata

Manuali da consegnare:

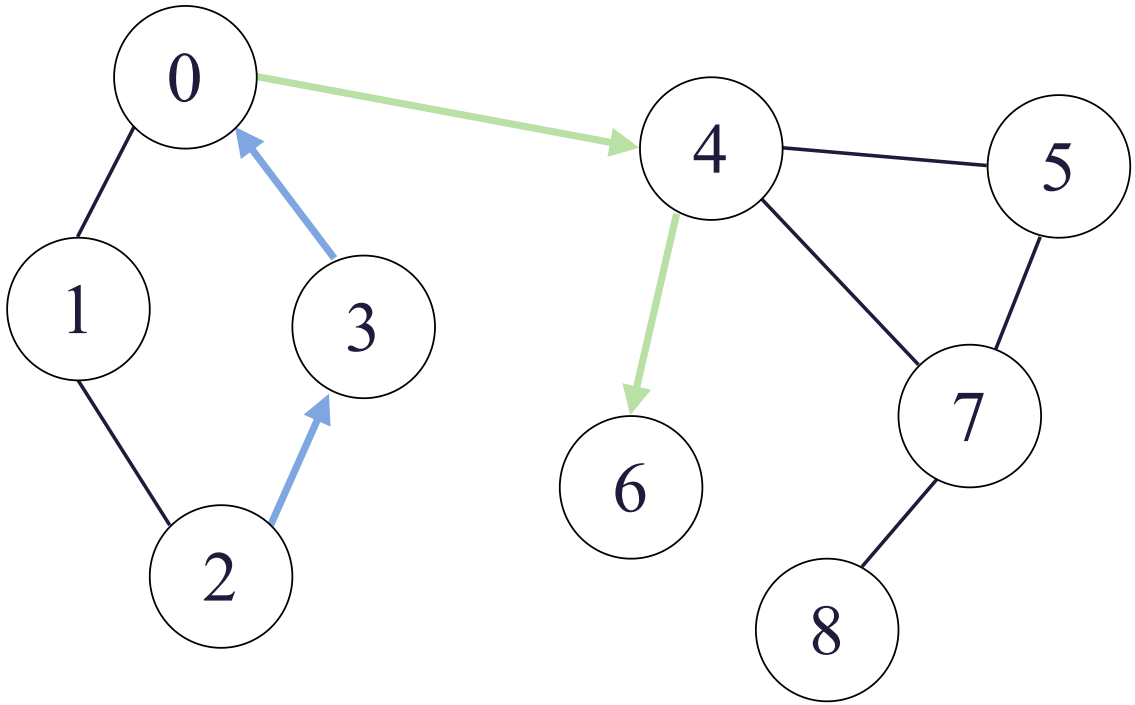
- $2 \to 6$
- $1 \to 7$
- $0 \to 2$



Esempio 3.1 - Risposta errata

Manuali da consegnare:

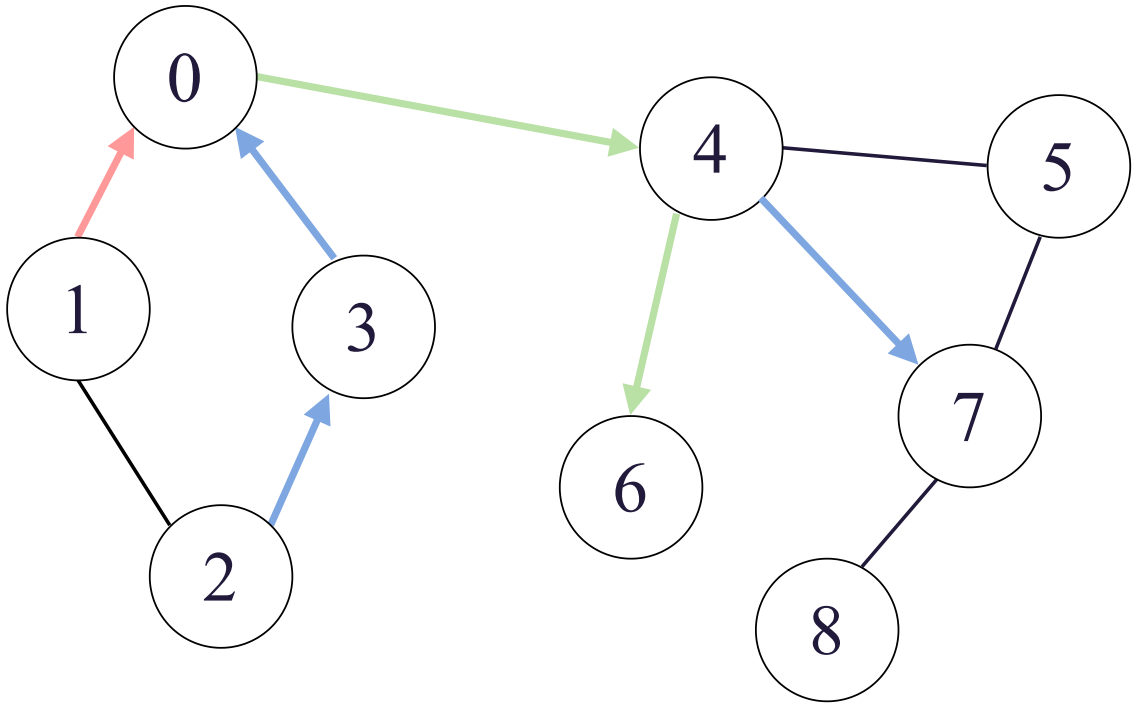
- $2 \to 6$ ✔
- $1 \to 7$
- $0 \to 2$



Esempio 3.2 - Risposta errata

Manuali da consegnare:

- $2 \to 6$ ✔
- $1 \to 7$ ✔
- $0 \to 2$



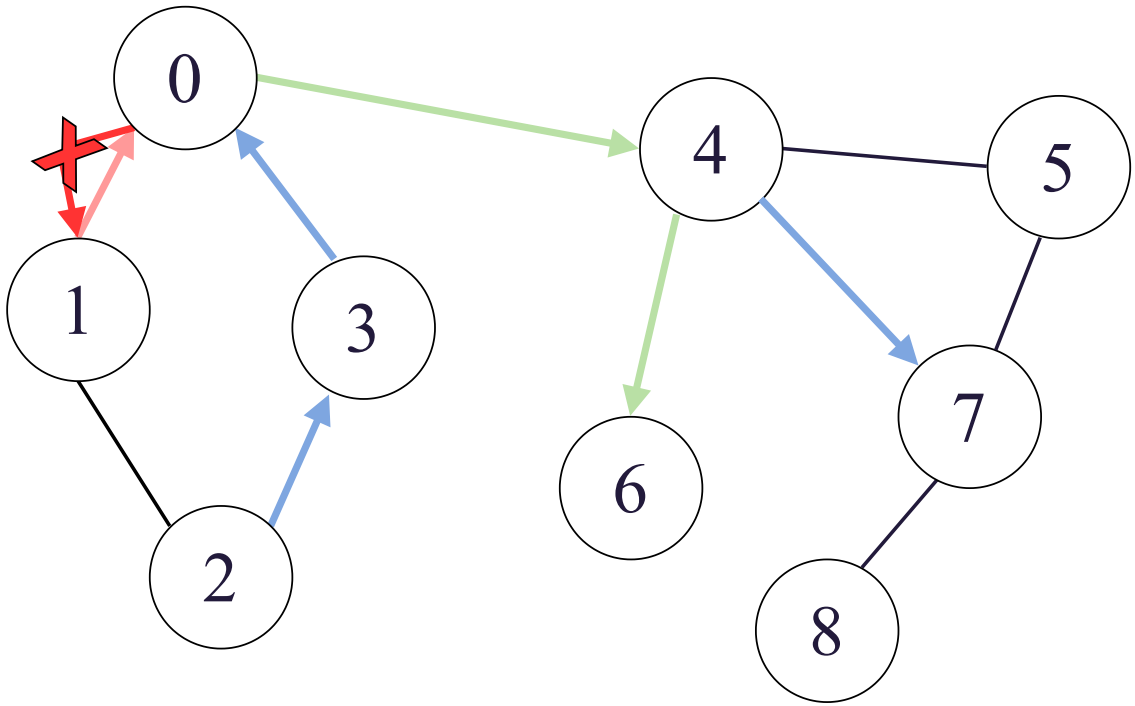
Esempio 3.3 - Risposta errata

Manuali da consegnare:

- $2 \to 6$ ✔
- $1 \to 7$ ✔
- $0 \to 2$ ✗

Dopo aver attraversato la strada $1 \to 0$ non è più possibile consegnare l'ultimo manuale!

Tuttavia, in questo caso **è possibile consegnare tutti i manuali** se si orientano le strade in modo intelligente.

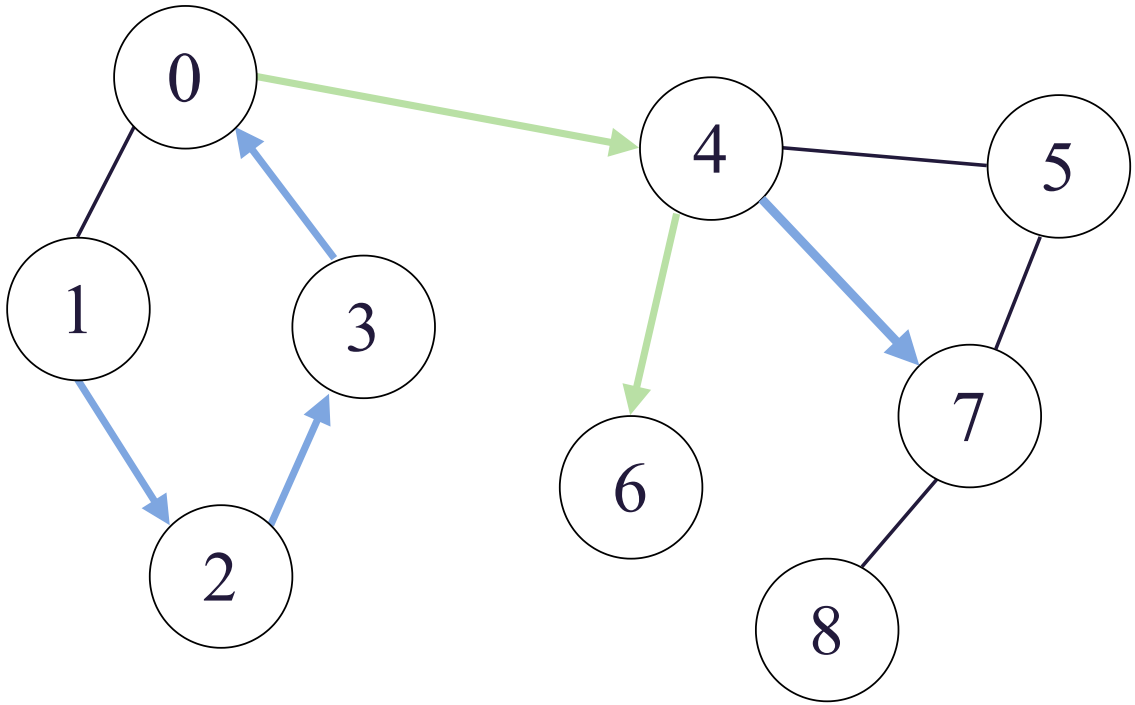


Esempio 3.4 - Risposta errata - soluzione

Manuali da consegnare:

- $2 \rightarrow 6$ ✔
- $1 \rightarrow 7$ ✔
- $0 \rightarrow 2$

Se per consegnare il secondo manuale avessimo orientato la strada da $1 \rightarrow 2$ invece che da $1 \rightarrow 0$ ...

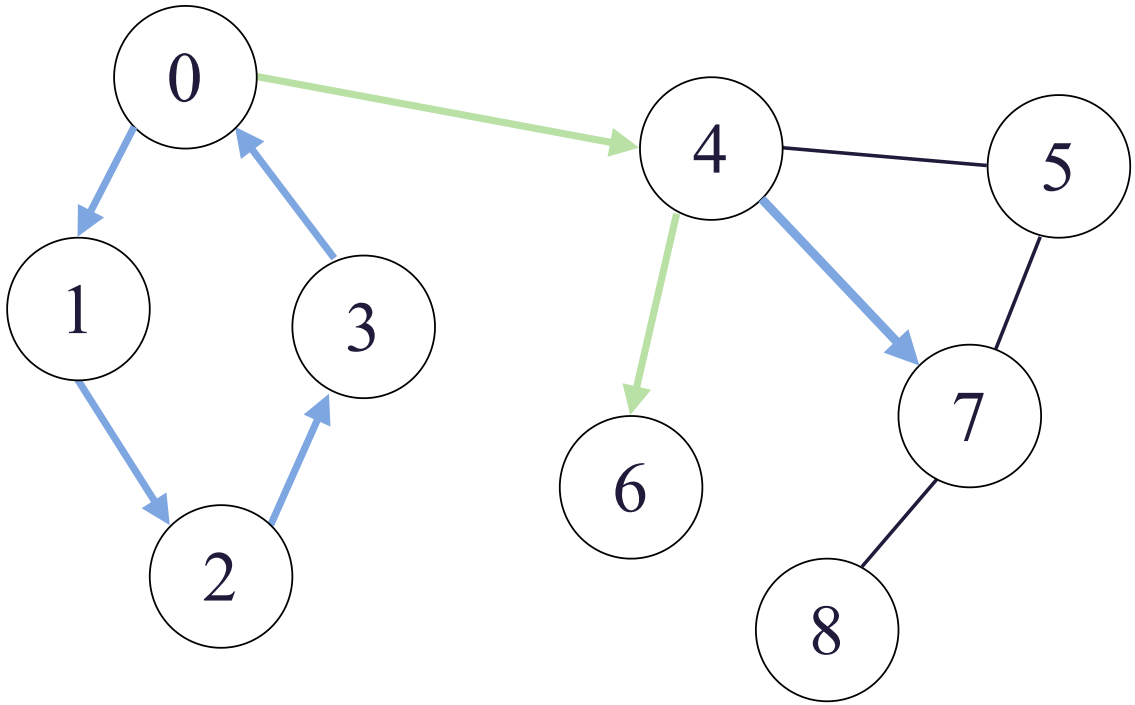


Esempio 3.5 - Risposta errata - soluzione

Manuali da consegnare:

- $2 \to 6$ ✔
- $1 \to 7$ ✔
- $0 \to 2$ ✔

Se per consegnare il secondo manuale avessimo orientato la strada da $1 \to 2$ invece che da $1 \to 0$ ...
Saremmo riusciti a risolvere il problema!



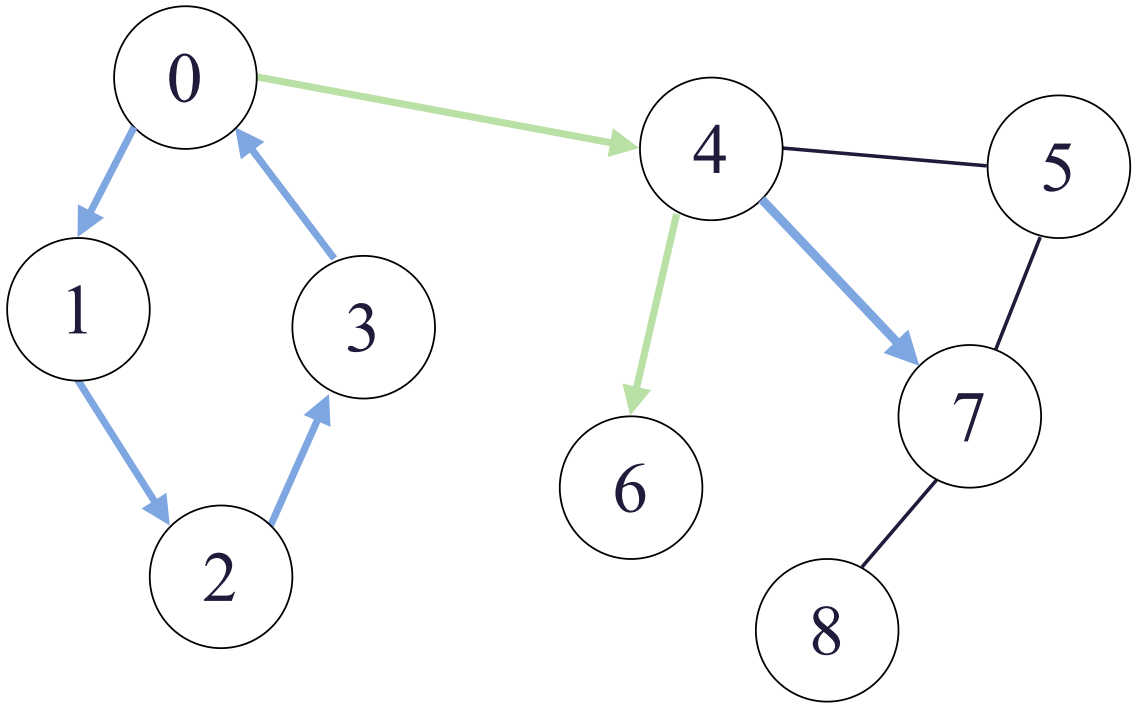
Esempio 3.6 - Risposta errata - soluzione

Manuali da consegnare:

- $2 \to 6$ ✔
- $1 \to 7$ ✔
- $0 \to 2$ ✔

In questo caso l'output corretto è il seguente:

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



Note su Input

Assunzioni generali

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

Casi di test

Ci sono 20 casi di test in totale:

- ☆★★ In 6 casi su 20, il grafo in input è **un albero e** $P = 1$;
- ☆☆★ In 9 casi su 20, il grafo in input **non è un albero**;
- ☆☆☆ In 5 casi su 20, il grafo in input è **un albero con** $N \geq 40000$ **e** $P \geq 50000$.

Per la sufficienza si possono risolvere i 6 casi di difficoltà ☆★★.

Limiti di tempo e memoria

I limiti di tempo e memoria sono:

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

$\Rightarrow$ Limite di 40 sottoposizioni per gruppo.

$\Rightarrow$ Potete provare con un dataset equivalente sulla vostra macchina (sito: https://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.

Punteggio

Ogni caso di test vale 5 punti. Il punteggio massimo è di 100 punti.

⇒ La **sufficienza è posta a 30 punti**.

- ✗ se la risposta che date **(possibile o impossibile)** non è corretta, si ottengono 0 punti.
- ✗ se la risposta corretta è **impossibile** e si stampa un arco che non genera conflitti, si ottengono 0 punti.
- ✗ se la risposta corretta è **possibile** e K non è corretto, si ottengono 0 punti.
- ✗ se la risposta corretta è **possibile** e la lista di archi orientati data in output non è corretta, si ottengono 0 punti.

Punti bonus per l'esame

L'assegnazione punti avviene in maniera competitiva:

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

Vengono considerati nella classifica per l'assegnazione dei punti solamente i **gruppi che raggiungono la sufficienza** (punteggio maggiore o uguale a 30).

$\Rightarrow$ Classifica:
https://asdlab.disi.unitn.it/arena/ranking/

Consegna

Consegna: venerdì 4 dicembre 2026 ore 18:00

Per caricare il vostro codice, recatevi su
`https://asdlab.disi.unitn.it/arena/`

Suggerimenti

Cominciate subito a lavorare al progetto per presentarvi al prossimo ricevimento (giovedì 9 dicembre) con tutte le domande che vorrete fare.
In ogni caso, sappiate che:

- potete venire a ricevimento
- rispondiamo su Telegram
- risponderemo alle vostre mail

Do's

È permesso:

- Discutere all'interno del gruppo
- Chiedere chiarimenti sul testo
- Chiedere opinioni su soluzioni
- Sfruttare codice fornito nei laboratori
- Utilizzare pseudocodice da libri o Wikipedia
- Richiedere aiuto (anche pesante) per la soluzione "minima"
- Venire a ricevimento
- Mandare meme (algoritmici) sul gruppo Telegram: it's your time to shine!

Don'ts

È vietato:

- Discutere con altri gruppi
- Mettere il proprio codice su repository pubblici
- Utilizzare codice scritto da altri
- Condividere codice (abbiamo potenti mezzi!)
- Chiedere suggerimenti online (es: stackoverflow)
- Fare riferimenti a soluzioni del progetto nei meme

Ricevimento

Date e orari

- lunedì 30 novembre 2026 dalle 16:00 alle 17:00 (online);
- mercoledì 2 dicembre 2026 (online o Povo 2, aula 178);
- giovedì 3 dicembre 2026 dalle 13:30 alle 15:30 (A101);
- venerdì 4 dicembre 2026 dalle 11:00 alle 12:00 (online);

$\Rightarrow$ Negli orari di ricevimento saremo a disposizione anche su Telegram. Se avete bisogno di un aiuto, scrivetelo sul gruppo del laboratorio.

$\Rightarrow$ Per qualsiasi domanda mandateci una mail a: `asd.disi@unitn.it` oppure contattateci su discord.