

Alpinocalypse Now (alpini)

Testo del problema

Slides originali su: judge.science.unitn.it/slides/asd17/prog3.pdf

Adunata degli Alpini Che grande orgoglio e che spettacolo! Quando gli alpini si riuniscono per l'adunata, migliaia di penne nere raggiungono la città prescelta arrivando da tutta Italia. Quest'anno siamo stati fieri di ospitare gli alpini proprio a Trento :)

La città in festa Nei giorni dell'adunata in città si fa festa! Lungo le strade si incontrano tanti alpini con sui si può cantare... o fare un brindisi!

La stanchezza L'ultimo giorno dell'adunata gli alpini sfilano per ore nelle vie principali. Poverini, i più anziani non hanno lo sprint di un tempo, tanto che i festeggiamenti prolungati possono risultare faticosi... molto faticosi!

Il vostro compito

Aiutate gli alpini “affaticati” (e forse rintronati dai festeggiamenti) a non stancarsi girando in tondo per le strade della città!

Bisogna chiudere alcuni incroci stradali, ma è bene chiuderne il **meno possibile** per non bloccare la città. Dopo aver chiuso alcuni incroci, non devono esistere percorsi ciclici in cui gli alpini possano girare in tondo stancandosi inutilmente.

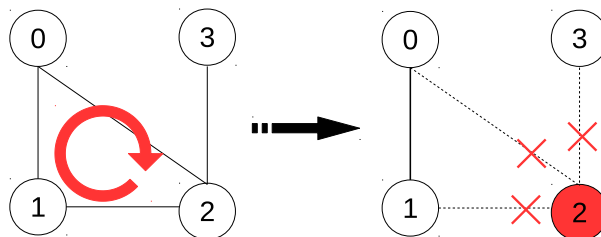
Input/Output

Input: un file contenente una lista di strade.

- La prima riga riporta 2 numeri, rispettivamente il numero di “incroci” e di “strade”;
- Ogni riga successiva è formata da 2 numeri e rappresenta una strada;
- Questi numeri identificano gli incroci collegati dalla strada in questione.

Output: Il numero di incroci da chiudere seguito dalla lista (possibilmente minimale) di incroci da chiudere per non avere percorsi ciclici.

Esempio



Punteggio

- 10 casi di test: ogni test assegna un punteggio (max 10 punti), che considera fino ai centesimi;
- Una soluzione proposta è valida se rende il grafo aciclico. Le risposte non valide non fanno punti!
- Le risposte valide invece ottengono questo punteggio:

$$P(S) = (1 - R/I) \times 10 + 0.1$$

NB: minore sarà il numero di nodi R rimossi, maggiore sarà il vostro punteggio!

- La sufficienza è posta a 1 punto. Vuol dire che avete perlomeno sempre fornito risposte valide;
- Non consideriamo soluzioni costanti. Non vale scrivere in output sempre la stessa cosa.

Note su input

Limiti generali Il numero di incroci (I) è sempre inferiore a 100:

- $3 \leq I < 100$

Il numero di strade S invece:

- In 3 casi di test su 20: $S \leq 100$
- In 6 casi di test su 20: $S \leq 500$
- In generale: $S \leq 2000$

Istruzioni per l'output

Se scrivete una soluzione esponenziale (tipo branch and bound) importate `alpini.h` (scaricabile da judge).

Man mano che migliorate la soluzione, scrivetela in output terminando la riga con `#`. La libreria arresterà il programma prima del timeout.

Il main va sempre dichiarato come `int main()` o `int main(void)`. Questo esercizio deve essere svolto in C++, non è possibile usare il C.

```
... include delle librerie di sistema ...
# include "alpini.h"

int main() {
    ...
}
```

Il correttore considererà l'ultima riga di output che finisce con `#`. Quindi, anche se non appendete soluzioni multiple, terminate l'output con `#`.

Comando di compilazione (sistemi Linux-based e Windows)

Supponendo che il sorgente con il vostro codice si chiami file `alpini.cpp`, i file `alpini.cpp`, `grader.cpp` e `tsp.h` devono essere nella stessa cartella e vengono compilati con il seguente comando:

```
/usr/bin/g++ -DEVAL -std=c++11 -O2 -pipe -static -s -o alpini grader.cpp alpini.cpp
```

Comando di compilazione (sistemi Mac OS X)

Su sistemi Mac OS X usate il seguente comando di compilazione:

```
/usr/bin/g++ -DEVAL -std=c++11 -O2 -pipe -o tsp alpini.cpp alpini.cpp
```

Se ottente un errore del tipo: `use of undeclared identifier quick_exit`, sostituite in `grader.cpp` l'istruzione `quick_exit(EXIT_SUCCESS);` con `exit(EXIT_SUCCESS);`.

Note su Output e dataset di esempio

Attenzione! Per gli input forniti nel dataset di esempio non è stata calcolata una soluzione ottima. Per questo motivo il dataset non contiene anche i relativi output, solitamente messi a disposizione.

Gli output mostrati nel resto del testo sono forniti a puro titolo illustrativo: rispettano la sintassi richiesta dal correttore ma non rappresentano necessariamente la soluzione migliore per il relativo input. Il secondo esempio mostra una soluzione in cui sono state prodotte più soluzioni, il correttore considera l'ultima riga terminata da #.

Esempi di input/output

File input.txt	File output.txt
4 4 0 1 1 2 2 0 2 3	1 2#
File input.txt	File output.txt
10 45 5 9 4 7 1 3 4 8 5 6 2 8 6 9 0 7 8 9 2 6 1 6 3 7 2 5 0 3 5 8 1 2 6 7 2 9 1 5 3 6 0 4 7 9 0 1 4 6 4 5 1 4 3 9 0 5 1 9 4 9 0 8 3 5 2 7 6 8 7 8 5 7 2 4 3 8 0 6 1 8 1 7 0 9 2 3 3 4 0 2	10 0 1 2 3 4 5 6 7 8 9# 9 0 1 2 3 4 5 6 7 8# 8 0 1 2 3 4 5 6 7#