

# PRIMO PROGETTO ASD 2019/2020



## Cicli ciclabili

# UN NUOVO SINDACO IN CITTÀ

A Trento è stato eletto un nuovo sindaco! È un temuto professore di algoritmi, che nel tempo libero ha deciso di dedicarsi alla politica. Con il partito «Algoritmi, Cricche e Libertà», ha stravinto le elezioni!



# IL SUO PROGRAMMA

Oltre ad avere a cuore i suoi studenti,  
il nuovo sindaco tiene molto alla città  
ed i suoi cittadini.

# IL SUO PROGRAMMA

Oltre ad avere a cuore i suoi studenti, il nuovo sindaco tiene molto alla città ed i suoi cittadini.

Dopo un consulto con l'assessore alla Salute (M. F.) e l'assessore allo Sport (M. B.), ha stilato una lista di interventi che vorrà fare per la città. Tra le varie cose, ha pensato di incentivare l'uso delle biciclette e costruire nuove piste ciclabili!

# IL SUO PROGRAMMA

Oltre ad avere a cuore i suoi studenti, il nuovo sindaco tiene molto alla città ed i suoi cittadini.

Dopo un consulto con l'assessore alla Salute (M. F.) e l'assessore allo Sport (M. B.), ha stilato una lista di interventi che vorrà fare per la città. Tra le varie cose, ha pensato di incentivare l'uso delle biciclette e costruire nuove piste ciclabili!

In realtà, stufo di trovare i 5 sempre troppo pieni, vorrebbe convincere gli studenti ad andare a Povo in bicicletta, per poi riuscire finalmente a trovare un posto sull'autobus!



Per attuare ciò, ha identificato  $N$  luoghi principali in tutta la città, punti di maggior interesse per gli abitanti. Le piste ciclabili già presenti e che verranno costruite sono tutte bidirezionali e collegano ciascuna due luoghi diversi. Come primo obiettivo, ha voluto assicurarsi che **tutti i luoghi siano sempre raggiungibili dovunque ci si trovi in città.**

Dopo un ulteriore consulto con il Vicesindaco (C. C.) - con delega al Riposo e al Divertimento - si decide di introdurre una miglioria alle piste che ridurrà la fatica degli studenti: ora nessuno dovrà più fare giri inutilmente!

- ▶ tutti i luoghi che fanno parte di un **percorso circolare semplice**, dovranno essere collegati tra loro in maniera diretta. Un percorso circolare semplice è un insieme di piste ciclabili che collega un punto della città a se stesso, senza passare per nessun luogo due volte.

In altre parole, **l'insieme di luoghi in ogni ciclo semplice, costituisce un sottografo completo.**



# IL VOSTRO COMPITO

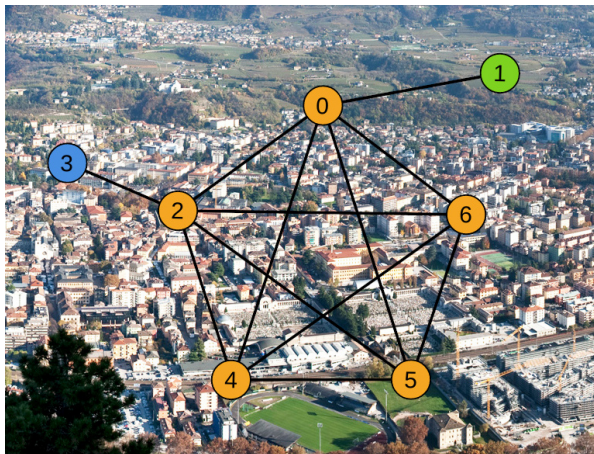
Dopo l'inaugurazione delle nuove piste ciclabili, vari abitanti si chiedono come raggiungere i vari punti della città nel modo più veloce possibile. Sono molte le domande che arrivano al sindaco, che avrà bisogno di aiuto per rispondere a tutti! Il sindaco vi fornirà:

- ▶ la mappa della città, con gli  $N$  luoghi di interesse e le piste ciclabili che ha fatto costruire, facendo rispettare i requisiti sopra elencati.
- ▶ un elenco di tutte le richieste dei cittadini, che indicheranno il punto da cui vogliono partire e quello che vogliono raggiungere.

## IL VOSTRO COMPITO

Per ciascuna richiesta dovrete comunicare la **minima distanza** che il cittadino dovrà percorrere, in termini di numero di piste ciclabili attraversate.

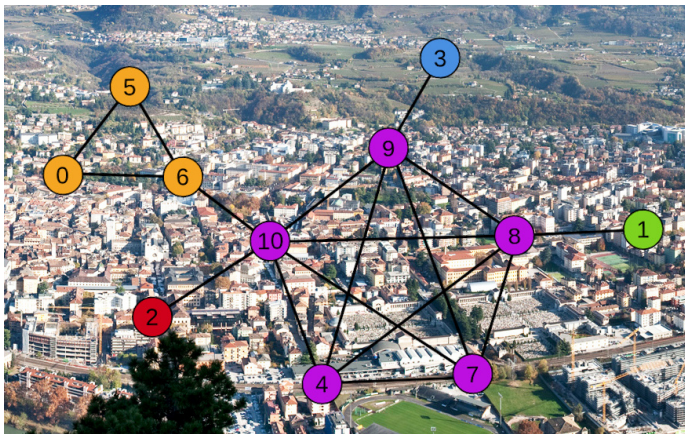
# ESEMPIO I



**Richieste** (partenza, destinazione): (0, 1), (6, 3), (3, 1), (6, 5), (3, 5).

**Risposte** (distanze minime): 1, 2, 3, 1, 2.

# ESEMPIO II

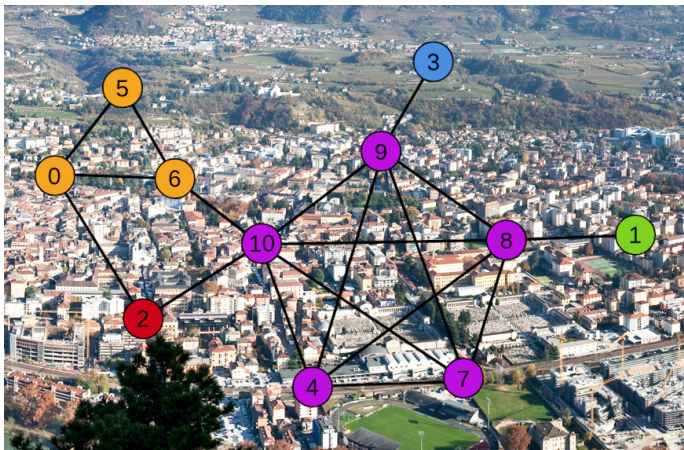


**Richieste** (partenza, destinazione): (9, 10), (1, 3), (2, 3), (9, 1), (10, 6).

**Risposte** (distanze minime): 1, 3, 3, 2, 1.

## ESEMPIO III - INPUT NON VALIDO

**Non** troverete tra gli input mappe come questa:



Infatti, il ciclo formato dai nodi 0, 2, 10, 6, 5 non costituisce un sottografo completo.

## Input:

- La prima riga riporta 3 numeri interi positivi:  $N$ ,  $M$  e  $Q$ , rispettivamente il numero di luoghi, di piste ciclabili e di richieste.
- Le successive  $M$  righe descrivono la mappa: ciascuna riga contiene due interi  $a_i$  e  $b_i$ , compresi tra 0 ed  $N - 1$ , ad indicare che  $a_i$  e  $b_i$  sono collegati da una pista ciclabile.
  - ▶ Nota: è garantito che la mappa sia un grafo connesso.
- Le successive  $Q$  righe forniscono le richieste: ciascuna contiene due interi  $u_j$  e  $v_j$ , compresi tra 0 ed  $N - 1$ , i luoghi tra i quali si vuole conoscere la distanza.

**Output:**  $Q$  righe, la  $j$ -esima riga deve contenere la risposta alla  $j$ -esima richiesta: ossia il minimo numero di piste ciclabili che bisogna percorrere per arrivare da  $u_j$  a  $v_j$ .

# ESEMPIO I (INPUT/OUTPUT)

Input:

7 12 5  
0 1  
0 2  
0 4  
0 5  
0 6  
2 3  
2 4  
2 5  
2 6  
4 5  
4 6  
5 6

0 1  
6 3  
3 1  
6 5  
3 5

Output:

1  
2  
3  
1  
2

# ESEMPIO II (INPUT/OUTPUT)

Input:

11 17 5  
0 5  
0 6  
1 8  
2 10  
3 9  
4 7  
4 8  
4 9  
4 10  
5 6  
6 10  
7 8  
7 9  
7 10  
8 9  
8 10  
9 10

9 10  
1 3  
2 3  
9 1  
10 6

Output:

1  
3  
3  
2  
1

## ASSUNZIONI GENERALI

- $1 \leq N \leq 50000$
- $1 \leq M \leq 500000$
- $1 \leq Q \leq 50000$
- Ogni grafo è connesso.
- Ogni grafo è non diretto.



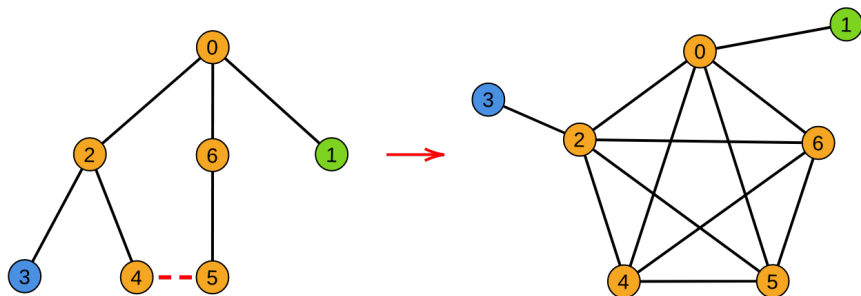
- Ci sono 20 casi di test in totale.
- In almeno 6 casi  $N \leq 10000$  e  $Q \leq 10000$ .
- In almeno 10 casi l'input è un **albero**.
- In almeno 14 casi l'input è un **albero con al più una cricca**.

I limiti di tempo e memoria sono:

- ▶ Tempo limite massimo: 2 secondi.
- ▶ Memoria massima: 32 MB.

## ESEMPIO (ALBERO CON UNA CRICCA)

Un albero con una cricca è costruito aggiungendo un arco tra due nodi in modo da creare un ciclo e collegando tutti i nodi che fanno parte del ciclo.



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

Per ogni caso di test per cui la vostra soluzione fornisce un output entro i limiti di tempo e memoria:

- se avete calcolato correttamente **tutte** le distanze richieste ottenete **5 punti**;
  - in caso contrario, **0 punti**.
- ✗ se uno o più risposte sono errate si ottengono comunque **0 punti**.
- ⇒ La **sufficienza è posta a 30 punti**.
- ⇒ C'è un limite di 40 sottoposizioni per gruppo.

L'assegnazione punti avviene in maniera competitiva:

- **3 punti** ai gruppi nel primo terzile della classifica (primo terzo della classifica, punteggio maggiore a quello fatto da almeno 2/3 dei gruppi);
- **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).

**Consegna: 13 dicembre ore 13:00**

Per caricare il vostro codice, recatevi su  
<https://judge.science.unitn.it/arena>

## SUGGERIMENTI

Cominciate subito a lavorare al progetto per presentarvi al prossimo laboratorio (martedì 10 dicembre) con tutte le domande che vorrete fare.

In ogni caso, sappiate che:

- potete venire a ricevimento
- risponderemo alle vostre mail

## È PERMESSO:

- 1 Discutere all'interno del gruppo
- 2 Chiedere chiarimenti sul testo
- 3 Chiedere opinioni su soluzioni
- 4 Sfruttare codice fornito nei laboratori
- 5 Utilizzare pseudocodice da libri o Wikipedia
- 6 Richiedere aiuto (anche pesante) per la soluzione "minima"
- 7 Venire a ricevimento

## È VIETATO:

- 1 Discutere con altri gruppi
- 2 Mettere il proprio codice su repository pubblici
- 3 Utilizzare codice scritto da altri
- 4 Condividere codice (abbiamo potenti mezzi!)

## DATE E ORARI

- lunedì 9 dicembre 2019 dalle 9:00 alle 11:00;
- martedì 10 dicembre 2019 dalle 15:30 alle 17:30 in A101 (ricevimento in lab);
- mercoledì 11 dicembre 2019 dalle 10:30 alle 12:30;
- giovedì 12 dicembre 2019 dalle 10:30 alle 12:30 in B107 (ricevimento in lab);

⇒ Prima di venire a ricevimento (escludendo quelli in laboratorio), è obbligatorio richiedere appuntamento via mail  
(`martin.brugnara@unitn.it`) e  
(`marta.fornasier@studenti.unitn.it`).