

# Quarto allenamento

Olimpiadi Italiane di Informatica - Selezione territoriale

---

Luca Chiodini

[luca@chiodini.org](mailto:luca@chiodini.org) - [l.chiodini@campus.unimib.it](mailto:l.chiodini@campus.unimib.it)

5 aprile 2018

1. Lettura e analisi di un problema
2. Spiegazione teorica
3. Soluzione

# Lettura e analisi di un problema

---

## “Ponti e isole” (seconda gara OIS 2015)

## “Ponti e isole” (seconda gara OIS 2015)

A seguito di un violento maremoto alcuni dei ponti che collegano le  $N$  isole dell'arcipelago Nowhere sono stati distrutti e il governo deve correre ai ripari per non lasciare che alcune isolette rimangano isolate e irraggiungibili.

Il governo dell'arcipelago Nowhere ha quindi assunto Giorgio per determinare quale è il minimo numero di ponti che è necessario costruire in aggiunta agli  $M$  rimasti affinché l'arcipelago sia di nuovo connesso, ovvero sia possibile da ogni isola raggiungere tutte le altre isole. Aiuta Giorgio a svolgere il suo compito!

## “Ponti e isole” (seconda gara OIS 2015)

Il file `input.txt` è composto da  $M + 1$  righe. La prima riga contiene i due interi  $N$  e  $M$ . Le successive  $M$  righe contengono due interi ciascuna, gli indici `da[i]`, `a[i]` delle isole collegate dall' $i$ -esimo ponte.

<code>input.txt</code>	<code>output.txt</code>
4 2	1
1 3	
3 2	

## “Ponti e isole” (seconda gara OIS 2015)

Il file `input.txt` è composto da  $M + 1$  righe. La prima riga contiene i due interi  $N$  e  $M$ . Le successive  $M$  righe contengono due interi ciascuna, gli indici  $da[i]$ ,  $a[i]$  delle isole collegate dall' $i$ -esimo ponte.

<code>input.txt</code>	<code>output.txt</code>
4 2 1 3 3 2	1
<code>input.txt</code>	<code>output.txt</code>
2 0	1

## “Ponti e isole” (seconda gara OIS 2015)

### Esempio

È possibile che la risposta al problema sia 0?



# “Ponti e isole” (seconda gara OIS 2015)

## Esempio

È possibile che la risposta al problema sia 0?

## Assunzioni

- $1 \leq N \leq 10\,000$ .
- $0 \leq M \leq 100\,000$ .
- $0 \leq \text{da}[i], \text{a}[i] < N$  per ogni  $i = 0 \dots M - 1$ .
- Per ogni coppia di isole esiste al più un ponte che le collega, e i ponti non vengono ripetuti nell'input.
- Nessun ponte collega un'isola a se stessa.
- Le isole sono numerate a partire da 0.
- Se l'arcipelago è già connesso, rispondere il valore 0.

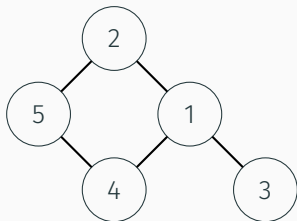
## Spiegazione teorica

---

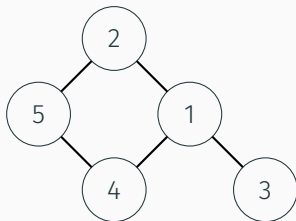
## Definizione

Un grafo è un insieme di elementi detti nodi (o vertici) che possono essere collegati fra loro da linee chiamate archi.

Formalmente, chiamiamo *grafo* una coppia ordinata  $G = (V, E)$  dove  $V$  è l'insieme dei nodi ed  $E$  è l'insieme degli archi, tali che  $E \subseteq V \times V$ .

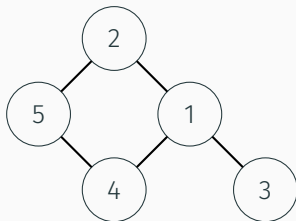


$V =$



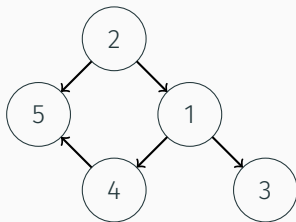
$$V = \{1, 2, 3, 4, 5\}$$

$$E =$$



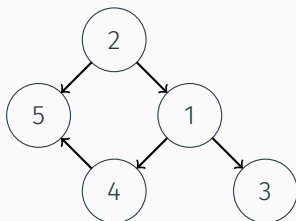
$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{(2, 1), (1, 4), (4, 5), (1, 3), (2, 5)\}$$



$$V = \{1, 2, 3, 4, 5\}$$

$$E =$$



$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{(2, 1), (1, 4), (4, 5), (1, 3), (2, 5)\}$$

Questa tipologia di grafo è detta **grafo orientato**: gli archi hanno una direzione e le coppie in  $E$  sono ordinate.

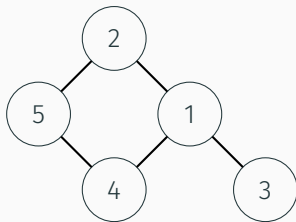


## Cammino

Un cammino dal nodo  $v_1$  al nodo  $v_n$  è una sequenza ordinata di vertici  $(v_1, v_2, \dots, v_n)$  tali che esistono gli archi  $(v_1, v_2), \dots, (v_{n-1}, v_n)$ .

## Cammino

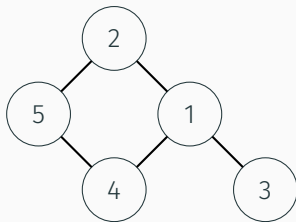
Un cammino dal nodo  $v_1$  al nodo  $v_n$  è una sequenza ordinata di vertici  $(v_1, v_2, \dots, v_n)$  tali che esistono gli archi  $(v_1, v_2), \dots, (v_{n-1}, v_n)$ .



# Caratteristiche dei grafi

## Cammino

Un cammino dal nodo  $v_1$  al nodo  $v_n$  è una sequenza ordinata di vertici  $(v_1, v_2, \dots, v_n)$  tali che esistono gli archi  $(v_1, v_2), \dots, (v_{n-1}, v_n)$ .



## Ciclo

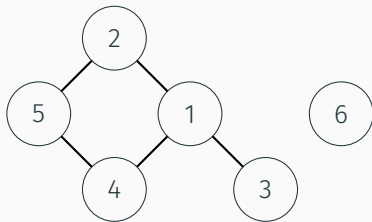
Un ciclo è un cammino chiuso, ovvero con  $v_1 = v_n$ .

## Raggiungibilità

Un nodo  $v$  è raggiungibile dal nodo  $u$  sse esiste un cammino tra  $u$  e  $v$ .

## Raggiungibilità

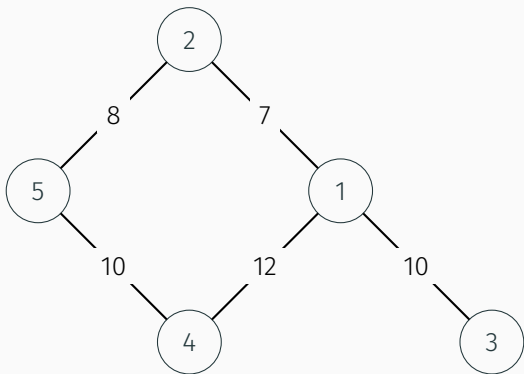
Un nodo  $v$  è raggiungibile dal nodo  $u$  sse esiste un cammino tra  $u$  e  $v$ .



# Caratteristiche dei grafi

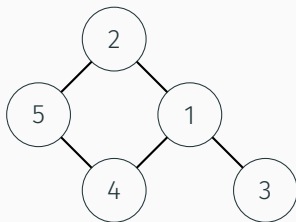
## Cammino minimo

Tra tutti i cammini esistenti tra  $u$  e  $v$ , chiamiamo *cammino minimo* quello che minimizza il costo di attraversamento degli archi che lo compongono.





# Rappresentazione di grafi



Matrice di adiacenza

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$m_{ij} = \begin{cases} 1 & \text{sse } (i,j) \in E \\ 0 & \text{altrimenti} \end{cases}$$



### Occupazione di memoria

Quanto occupa una matrice di adiacenza per rappresentare un grafo (ricordiamo  $G = (V, E)$ )?

# Rappresentazione di grafi - Matrici di adiacenza

## Occupazione di memoria

Quanto occupa una matrice di adiacenza per rappresentare un grafo (ricordiamo  $G = (V, E)$ )?

$$\mathcal{O}(|V|^2)$$

Intuitivamente, dov'è lo “spreco” di spazio?

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

# Rappresentazione di grafi - Matrici di adiacenza

## Occupazione di memoria

Quanto occupa una matrice di adiacenza per rappresentare un grafo (ricordiamo  $G = (V, E)$ )?

$$\mathcal{O}(|V|^2)$$

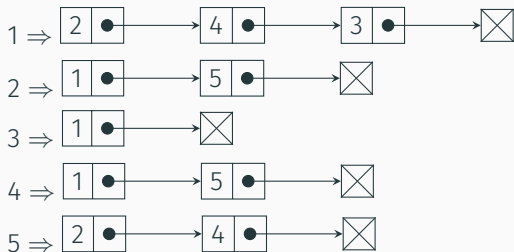
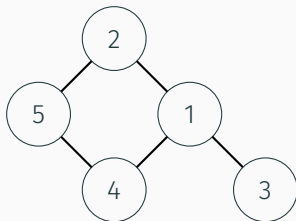
Intuitivamente, dov'è lo “spreco” di spazio?

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Stiamo memorizzando tantissimi “zeri”: la matrice di adiacenza consuma troppo spazio soprattutto per grafi *sparsi*.



## Rappresentazione di grafi - Liste di adiacenza



```
#include <list>
using namespace std;

#define MAXN 10000

list<int> grafo[MAXN];

for (int i = 0; i < M; i++)
{
    int u = da[i], v = a[i];
    grafo[u].push_back(v);
    grafo[v].push_back(u);
}
```

## Algoritmi di visita di un grafo

Dato un grafo vorremmo essere in grado, ovviamente con un algoritmo, di “visitare” (ovvero scoprire, enumerare, cercare) tutti i suoi vertici.

# Algoritmi di visita di un grafo

Dato un grafo vorremmo essere in grado, ovviamente con un algoritmo, di “visitare” (ovvero scoprire, enumerare, cercare) tutti i suoi vertici.

## Visita in profondità (DFS)

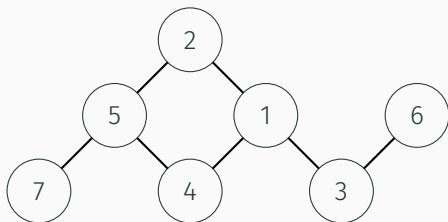
Per ogni nodo, visita fin quando possibile uno dei nodi ad esso adiacente.

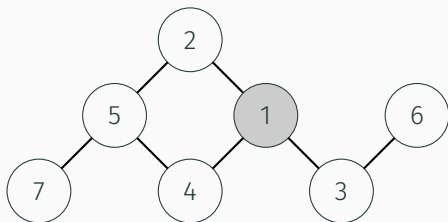
## Visita in ampiezza (BFS)

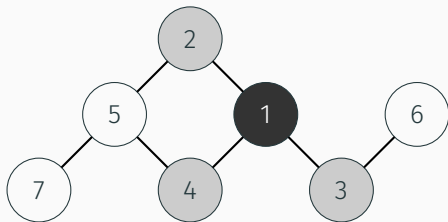
Per ogni nodo, visita tutti i nodi ad esso adiacenti prima di spostarsi agli adiacenti degli adiacenti.

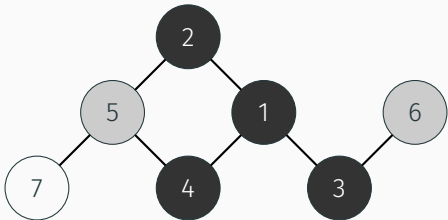
<https://visualgo.net/en/dfsdfs>

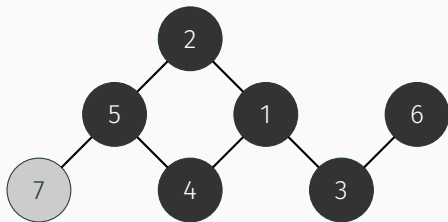


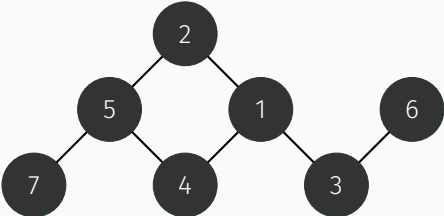


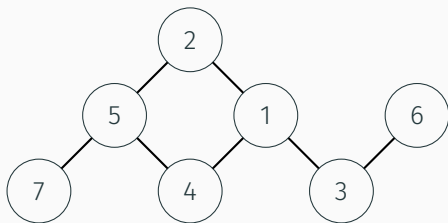


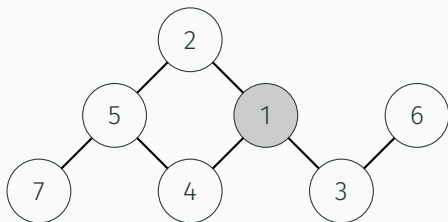




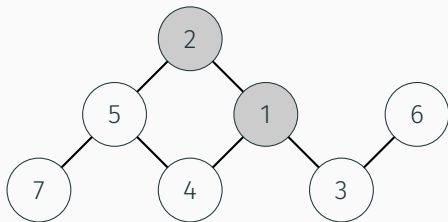


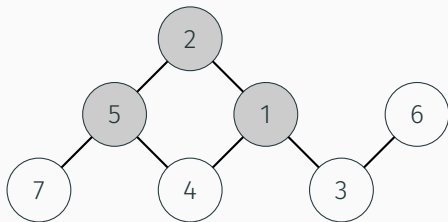


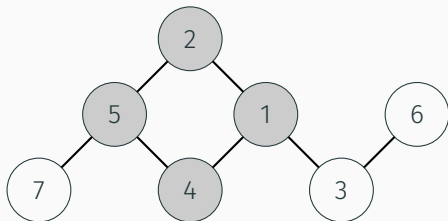


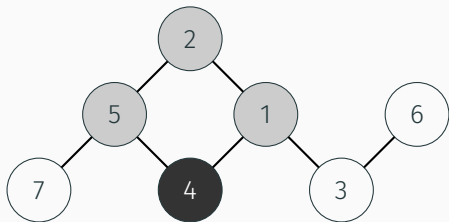


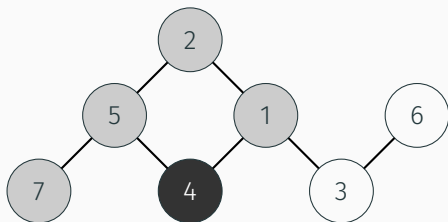


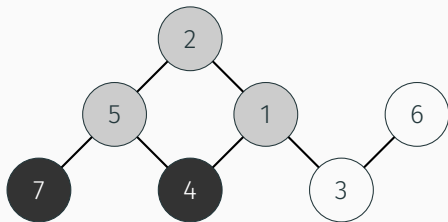


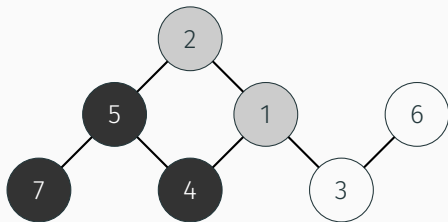


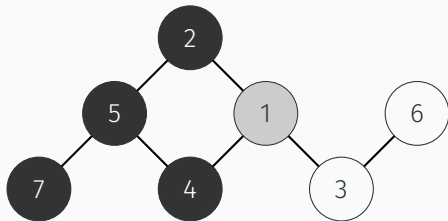




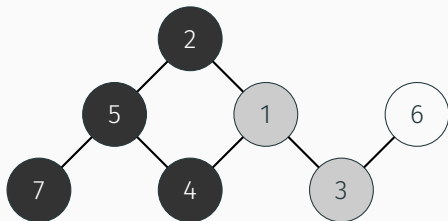


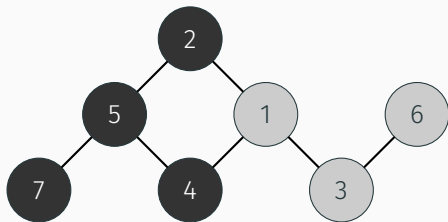


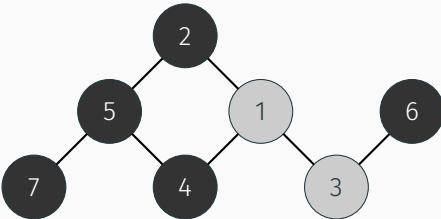


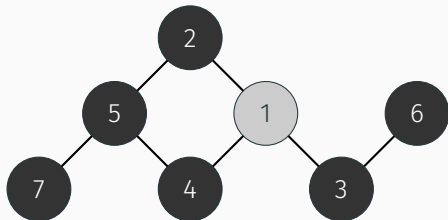


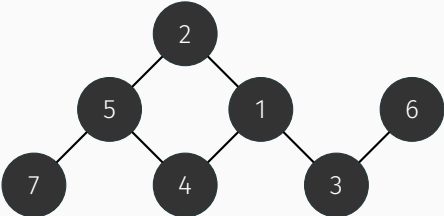












```
void bfs(int nodo_partenza)
{
    queue<int> coda;
    coda.push(nodo_partenza);

    while (!coda.empty()) {
        int nodo = coda.front();
        coda.pop();
        visitato[nodo] = true;
        for (int adiacente: grafo[nodo])
            if (!visitato[adiacente])
                coda.push(adiacente);
    }
}
```

```
void dfs(int nodo)
{
    visitato[nodo] = true;

    // Scorre la lista di adiacenza di 'nodo'
    for (int adiacente: grafo[nodo])
        if (!visitato[adiacente])
            dfs(adiacente);
}
```

Soluzione

---



## “Ponti e isole” (seconda gara OIS 2015)

Ripropongo il testo. È più facile da leggere ora?

## “Ponti e isole” (seconda gara OIS 2015)

Ripropongo il testo. È più facile da leggere ora?

A seguito di un violento maremoto alcuni dei ponti che collegano le  $N$  isole dell'arcipelago Nowhere sono stati distrutti e il governo deve correre ai ripari per non lasciare che alcune isolette rimangano isolate e irraggiungibili.

Il governo dell'arcipelago Nowhere ha quindi assunto Giorgio per determinare quale è il minimo numero di ponti che è necessario costruire in aggiunta agli  $M$  rimasti affinché l'arcipelago sia di nuovo connesso, ovvero sia possibile da ogni isola raggiungere tutte le altre isole. Aiuta Giorgio a svolgere il suo compito!

## Modellazione tramite grafo

---

input.txt	output.txt
7 4	2
1 4	
2 5	
1 3	
3 6	

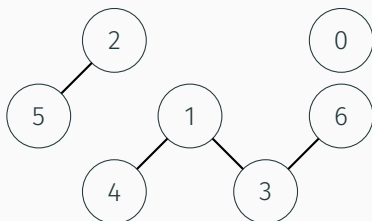
---

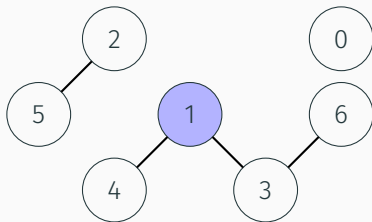
# Modellazione tramite grafo

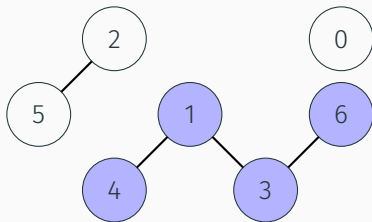
---

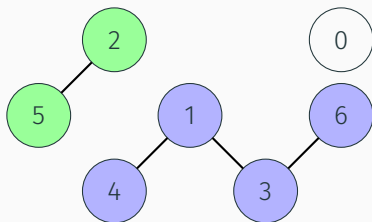
input.txt	output.txt
7 4	2
1 4	
2 5	
1 3	
3 6	

---

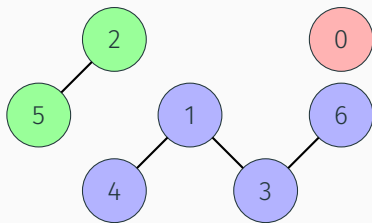






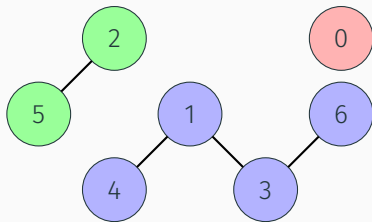


# Idea di soluzione





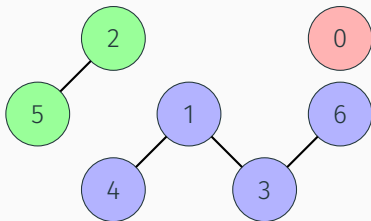
# Idea di soluzione



## Soluzione

Qual è la risposta al problema?

# Idea di soluzione



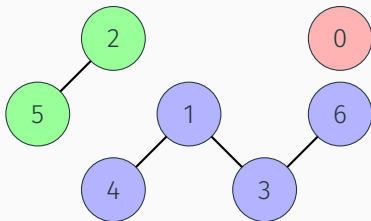
## Soluzione

Qual è la risposta al problema? “Numero di colori” - 1

## Complessità computazionale

Qual è la complessità computazionale della soluzione?

# Idea di soluzione



## Soluzione

Qual è la risposta al problema? “Numero di colori” - 1

## Complessità computazionale

Qual è la complessità computazionale della soluzione?

Ciascuna visita impiega  $\mathcal{O}(N)$ , quindi sembrerebbe  $\mathcal{O}(N^2)$ . Tuttavia è semplice osservare che non visitiamo mai un nodo più di una volta, il che implica  $\mathcal{O}(N)$ .

Problemi su grafi... non troppo complessi

## Problemi su grafi... non troppo complessi

- depura
  - Territoriali 2009
  - Grafo delle precedenze
  - Visita del grafo

## Problemi su grafi... non troppo complessi

- **depura**
  - Territoriali 2009
  - Grafo delle precedenze
  - Visita del grafo
- **fulcro**
  - GATOR 2014
  - Visita del grafo

## Problemi su grafi... non troppo complessi

- **depura**
  - Territoriali 2009
  - Grafo delle precedenze
  - Visita del grafo
- **fulcro**
  - GATOR 2014
  - Visita del grafo
- **sentieri**
  - Territoriali 2016
  - 01-BFS oppure Dijkstra

## Esercizio

Implementare una soluzione per il problema “ponti”.

## Riferimenti

Questa presentazione e soluzione in C++ dell’esercizio:

[https://files.chiodini.org/OII\\_Territoriali\\_2018/](https://files.chiodini.org/OII_Territoriali_2018/)

- Piattaforma di allenamento con correttore e forum:

<https://training.olinfo.it>

- Guida alle selezioni territoriali del prof. Bugatti:

[http://www.imparando.net/sito/olimpiadi\\_di\\_informatica.htm](http://www.imparando.net/sito/olimpiadi_di_informatica.htm)