

ASD Laboratorio 03

The A(SD)-Team

UniTN

2026-10-29

CALENDARIO

24/09	Introduzione
22/10	Ad-hoc
29/10	Grafi 1
19/11	Grafi 2
26/11	Grafi 3
03/12	Lab Progetto 1

Dato un grafo **DIRETTO** $G(V, E)$, esso può essere rappresentato in vari modi¹:

- ❶ Liste di adiacenza: per ogni nodo $v \in V$, lista dei vicini;
- ❷ Matrice di adiacenza: $M[v][w] = 1$ se esiste l'arco $(v, w) \in E$, 0 altrimenti;
- ❸ Lista di archi: lista degli archi $(v, w) \in E$

⇒ se il grafo è **INDIRETTO** per ogni (v, w) aggiungiamo anche (w, v) .

Esempi scaricabili dal sito:

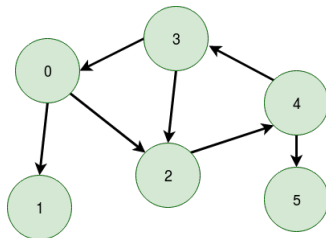
<https://asdlab.science.unitn.it/asd25/graphs.zip>.

¹([link di approfondimento](#))

FORMATO DI INPUT: ESEMPIO (I)

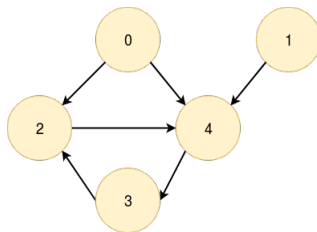
- riga 1: due interi N e M , rispettivamente il numero di nodi e di archi;
- righe 2 ... $M+1$: due interi s_i, t_i che denotano l'arco (s_i, t_i) . Eventualmente possono esserci altri attributi, per esempio un peso w_i .

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

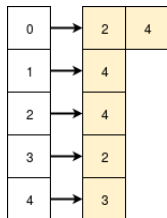


FORMATO DI INPUT: ESEMPIO (II)

5 6
0 2
0 4
1 4
3 2
2 4
4 3



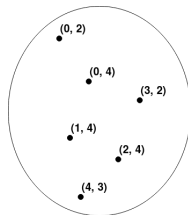
Liste di adiacenza:



Matrice di adiacenza:

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

Lista di archi:



LISTE DI ADIACENZA

ARRAY DI VECTOR

Lista di adiacenza implementata come array di vector o vector di vector. Se servono altre variabili (open, close) si creano altri array.

```
vector< vector <int> > adj;  
vector<bool> visitato;
```

ARRAY DI STRUCT

Struct nodo contenente lista di adiacenza e altre informazioni.

```
struct nodo{  
    vector<int> vic;  
    bool visitato= false;  
};
```

LISTE DI ADIACENZA

```
struct nodo{  
    vector<int> vic;  
    bool visitato= false;  
};  
vector<nodo> grafo;  
...  
in >> N;  
grafo.resize(N);  
for(int i=0;i<M;i++){  
    int from, to;  
    in >> from >> to;  
    grafo[from].vic.push_back(to);  
}
```

PROBLEMI

VISITA DI UN ALBERO

Dato un albero T radicato in R , visitare l'albero in pre-order, in-order e post-order. Questo task ha 20 casi di test, di cui 6 sono alberi binari.

VISITA DI GRAFO ORIENTATO

Dato un grafo ed un nodo di partenza, trovare il numero di nodi raggiungibili a partire da quel nodo.

DIAMETRO SU GRAFO NON ORIENTATO

Dato un grafo non orientato trovare il cammino minimo di lunghezza massima tra ogni coppia di nodi del grafo, ovvero il più lungo percorso minimo fra due nodi.

PROBLEMI

PRIMO PROGETTO A. A. 2016/2017

Pokémon Park

PRIMO PROGETTO A. A. 2024/2025

The Shining - Algoritmi alla follia