Grafi

Palestra di algoritmi

15 gennaio 2019

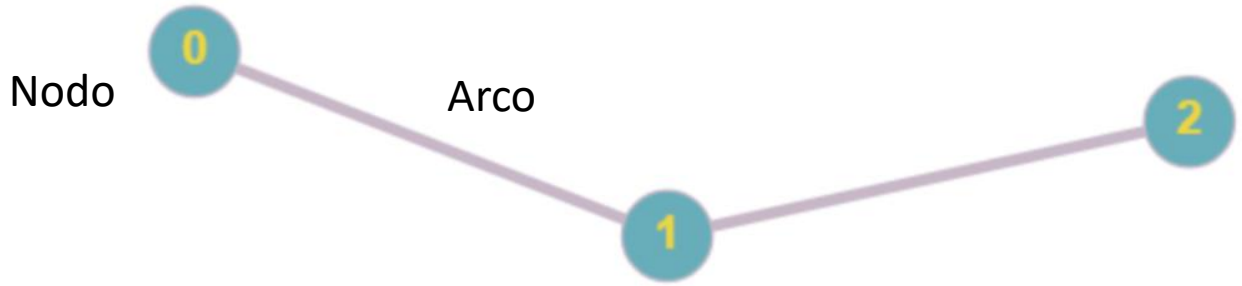
L’ABC – Cos’è un grafo?

Un grafo è una **struttura dati** composta di **nodi** (o vertici) e **archi.**

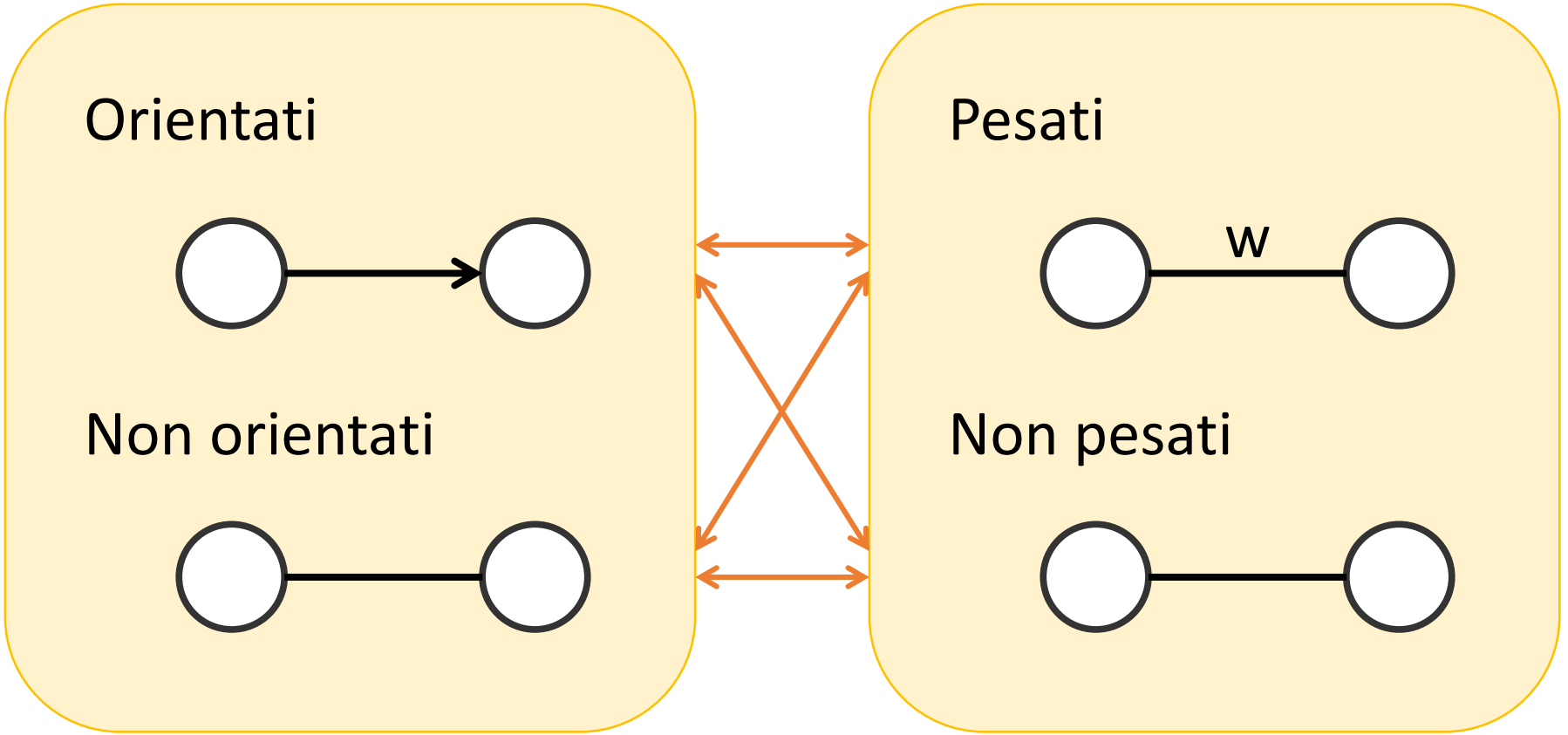
$$G = (V, E)$$

Insieme dei nodi — Insieme degli archi

Rappresentazione grafica:



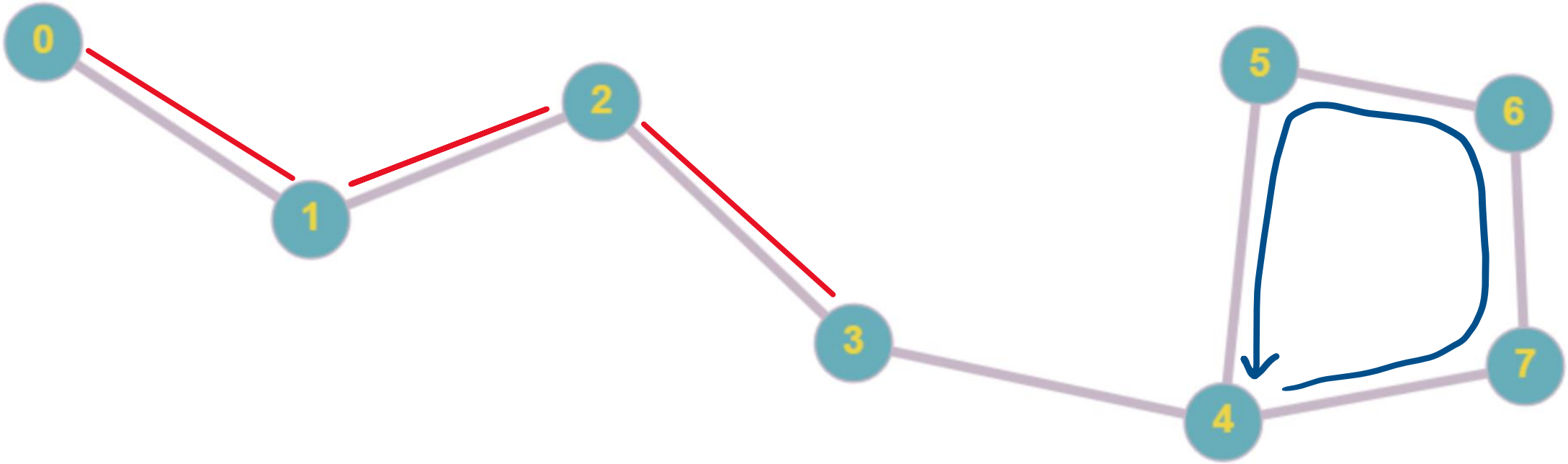
L’ABC – Gli archi



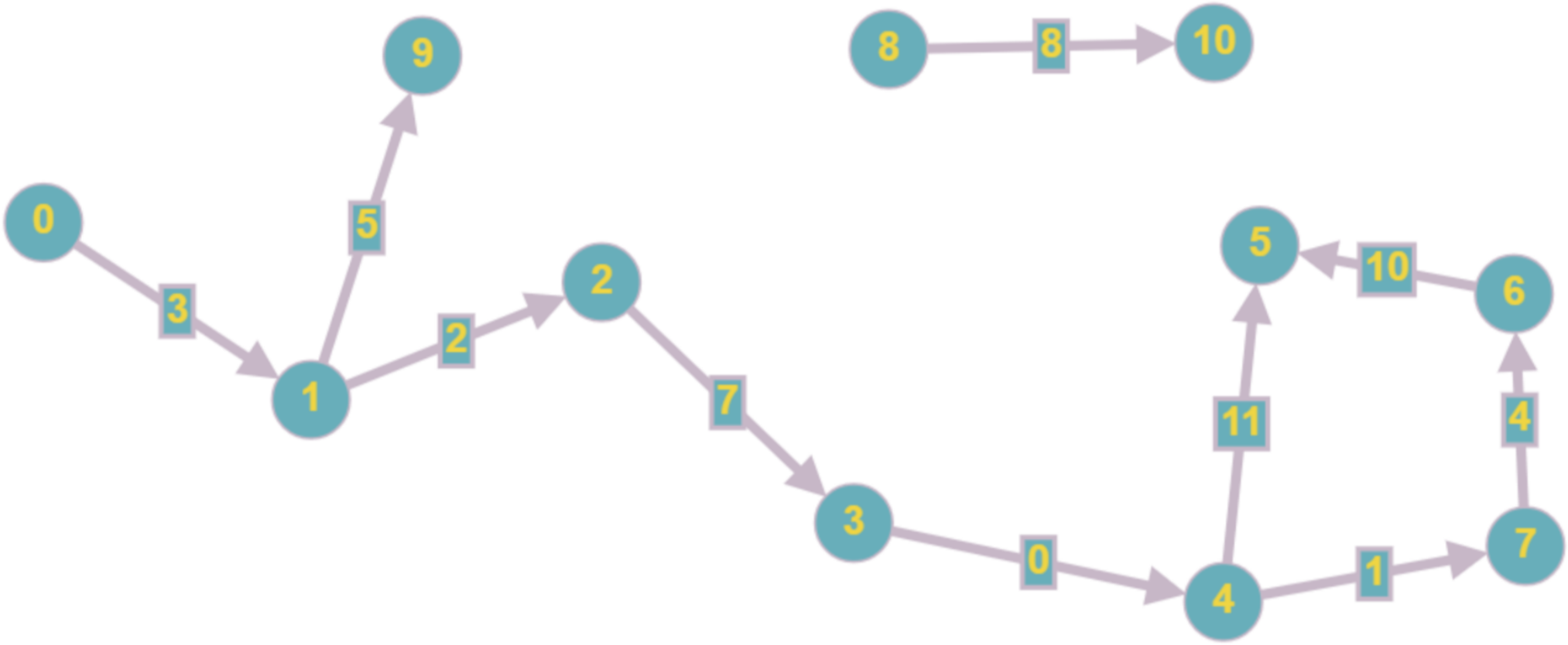
Posso trasformare un grafo non orientato in un grafo orientato? E viceversa?

L’ABC – Cammini e cicli

- Una successione di archi tra nodi adiacenti si chiama **cammino**.
- Un cammino che torna al suo punto di partenza è un **ciclo**.



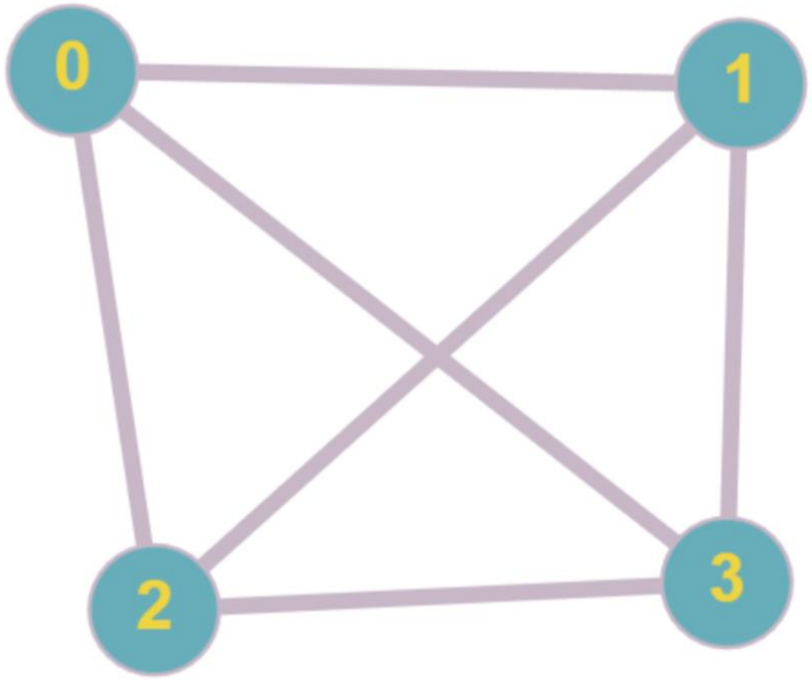
Esempio – Grafo orientato pesato



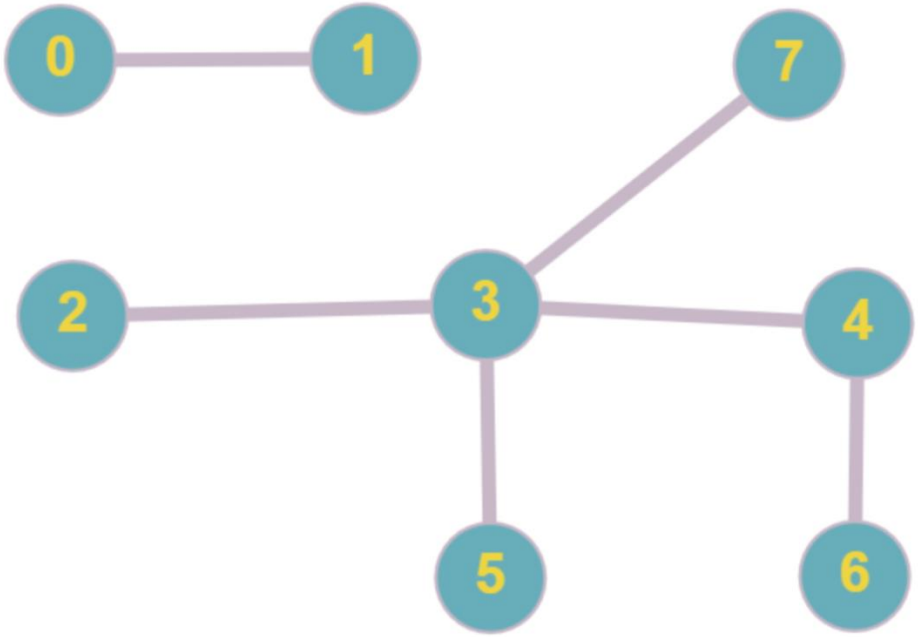
Casi particolari

Che cos’hanno questi grafi di particolare?

Grafo completo



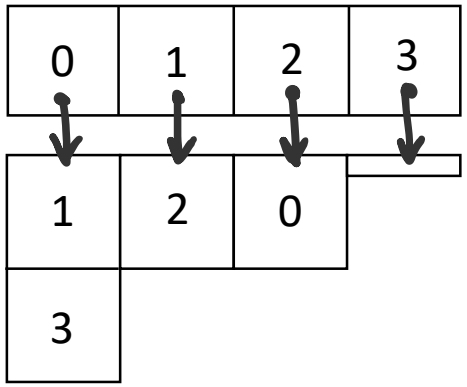
Foreste e alberi (radicati e non)

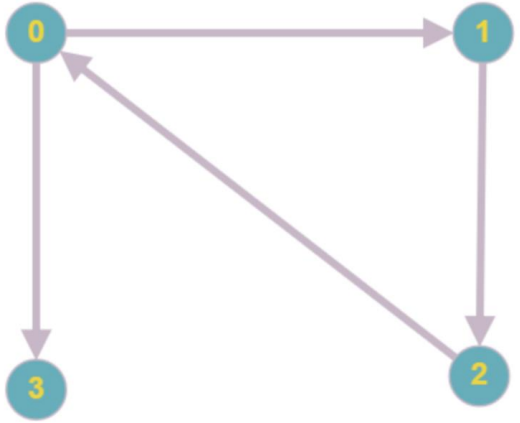


Memorizzare i grafi – liste di adiacenza

- Ogni nodo è rappresentato da un numero intero nell'insieme $[0..n-1]$, dove n è il numero di nodi.
- Per ogni nodo, salviamo tutti i suoi vicini in una lista (ad esempio un array dinamico – `std::vector`)

$\Longrightarrow$ Rappresenteremo il grafo come un `vector<vector<int> >`



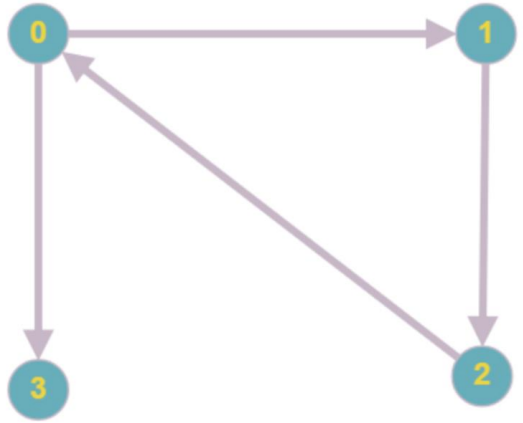


Come salvo un grafo pesato? E un grafo non orientato?

Memorizzare i grafi – Matrice di adiacenza

- Ogni nodo è rappresentato da un numero intero nell'insieme $[0..n-1]$, dove n è il numero di nodi.
- Per ogni coppia di nodi (i, j), indichiamo nella matrice se esiste un nodo da i a j

	0	1	2	3
0	0	1	0	1
1	0	0	1	0
2	1	0	0	0
3	0	0	0	0



Come salvo un grafo pesato? E un grafo non orientato?

Visite

Dato un nodo sorgente r visitare una ed una sola volta tutti i nodi che sono raggiungibili da r (ovvero visitare tutti i nodi u tali che esiste un cammino da r a u).

Visite

Dato un nodo sorgente r visitare una ed una sola volta tutti i nodi che sono raggiungibili da r (ovvero visitare tutti i nodi u tali che esiste un cammino da r a u).

Va bene un qualunque ordine di visita?

Visite

Dato un nodo sorgente r visitare una ed una sola volta tutti i nodi che sono raggiungibili da r (ovvero visitare tutti i nodi u tali che esiste un cammino da r a u).

Va bene un qualunque ordine di visita?

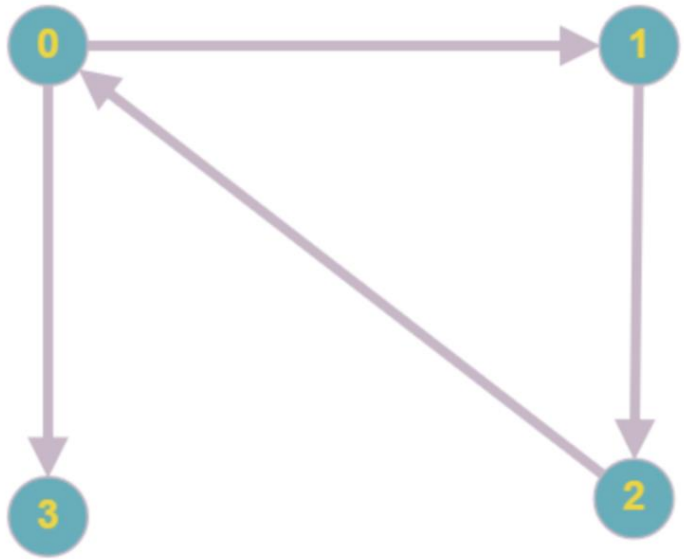
Ovviamente no! Ci sono algoritmi anche per questo

Depth first search

L’idea è quella di visitare completamente ogni *ramo* del grafo che parte dalla sorgente…

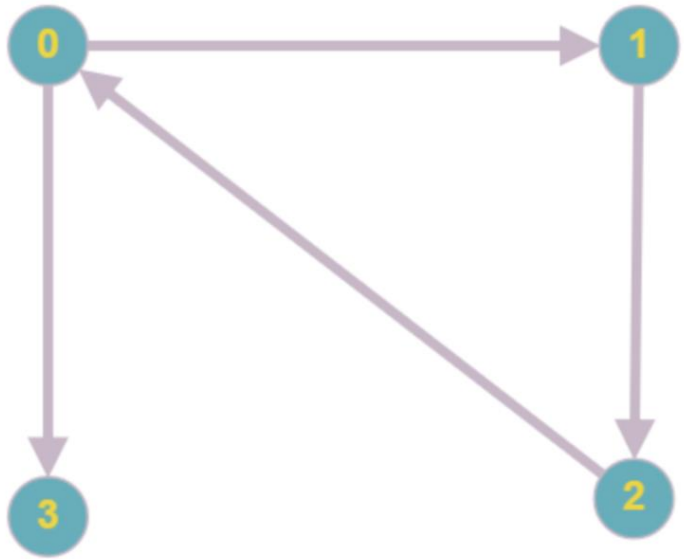
Depth first search

L’idea è quella di visitare completamente ogni *ramo* del grafo che parte dalla sorgente…



Depth first search

L’idea è quella di visitare completamente ogni *ramo* del grafo che parte dalla sorgente…



…In modo **ricorsivo**

Un esempio un po’ più elaborato (4.9)

Breadth first search e cammini minimi

L’idea è di visitare per primi tutti i nodi adiacenti alla sorgente, per poi estenderla ai vicini dei vicini e così via…

Ci servono due strutture dati: una per tener conto dei nodi già visitati, una per tener conto dei prossimi nodi da visitare.

Breadth first search e cammini minimi

L'idea è di visitare per primi tutti i nodi adiacenti alla sorgente, per poi estenderla ai vicini dei vicini e così via…

Ci servono due strutture dati: una per tener conto dei nodi già visitati, una per tener conto dei prossimi nodi da visitare.

Un array di booleani `visited` di dimensione n

Una coda di interi che contiene i prossimi nodi per la visita.