

# *ASD Laboratorio 05*

The A(SD)-Team

UniTN

2025-12-02

# CALENDARIO

09/10	Introduzione
21/10	Ad-hoc
04/11	Grafi 1
13/11	Grafi 2
02/12	Grafi 3
09/12	Lab Progetto 1

# MINI-PROGETTO!

## Mini-progetto **facoltativo**

- Dal 9 al 16 dicembre (consegna ore 18:00);
- Iscrizione dei gruppi al progetto entro **lunedì 8 dicembre**:  
[https://bit.ly/ASDprog\\_2025-2026](https://bit.ly/ASDprog_2025-2026) (dovete essere loggati con l'account UniTN)
  - ⇒ I gruppi valgono solo per questo progetto, ci sarà un'iscrizione a parte per il progetto del secondo semestre.

# SOLUZIONI: ALBERI RED-BLACK (RED-BLACK)

Fate riferimento alla slide del prof. Montresor sulle visite di alberi:

<http://disi.unitn.it/~montreso/asd/slides/06-abr.pdf>.

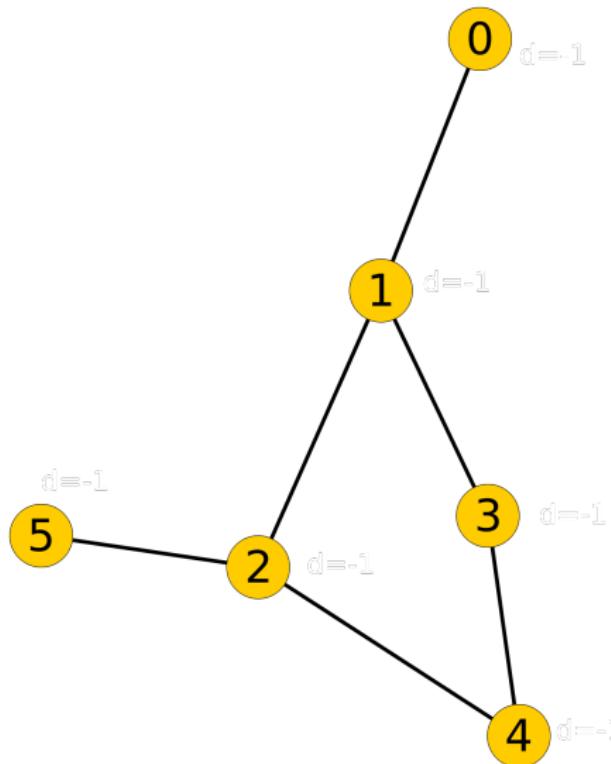
Gli alberi Red-Black sono alberi binari di ricerca in cui:

- Ogni nodo è colorato di rosso o di nero
- Le chiavi vengono mantenute solo nei nodi interni dell'albero
- Le foglie sono costituite da nodi speciali *Nil*
- Vengono rispettati i seguenti vincoli:
  - ① La radice è nera
  - ② Tutte le foglie sono nere
  - ③ Entrambi i figli di un nodo rosso sono neri
  - ④ Ogni cammino semplice da un nodo *u* ad una delle foglie contenute nel suo sottoalbero ha lo stesso numero di nodi neri

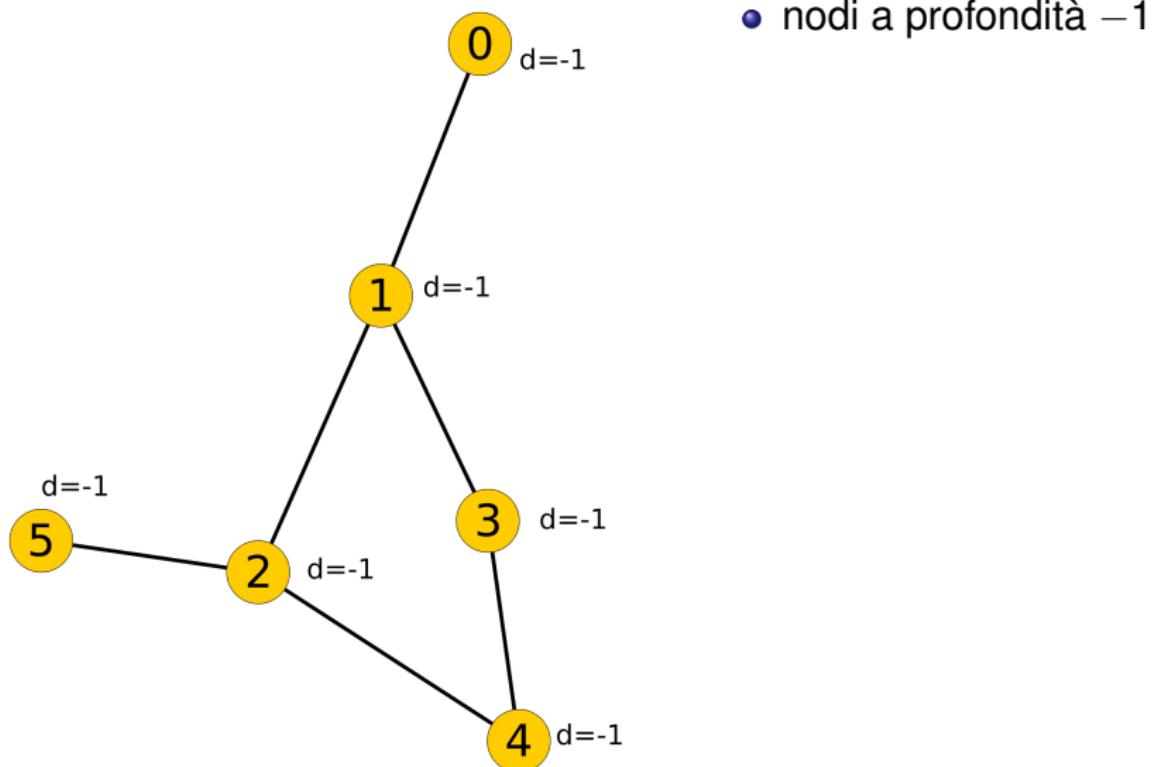
## SOLUZIONI: INDIVIDUAZIONE DI UN CICLO IN UN GRAFO ORIENTATO (CICLO)

- Individuiamo il ciclo e la sua lunghezza →  $C$ ;
  - Usiamo una visita in profondità e marchiamo ogni nodo visitato con la sua profondità rispetto alla sorgente;
  - Se incontriamo un nodo già visitato, che non sia il nodo da cui proveniamo (**padre** del nodo corrente rispetto alla visita):  
    ⇒ **ciclo**
  - La dimensione del ciclo è data dalla differenza di profondità fra i nodi collegati dall'arco che chiude il ciclo;
  - La visita va eseguita su tutti i nodi non visitati per tenere conto del caso in cui ci siano più componenti connesse.
- ⇒ complessità:  $\Omega(V + E)$

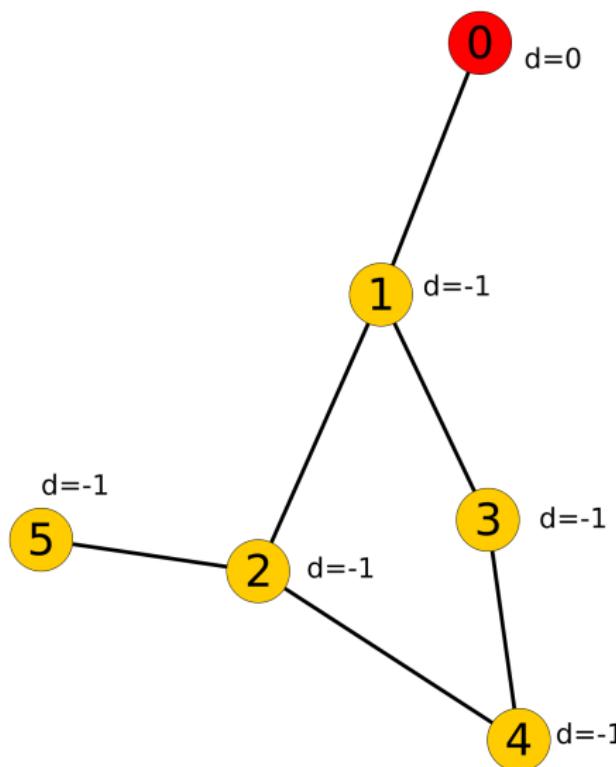
# INDIVIDUAZIONE DI UN CICLO (I)



# INDIVIDUAZIONE DI UN CICLO (I)

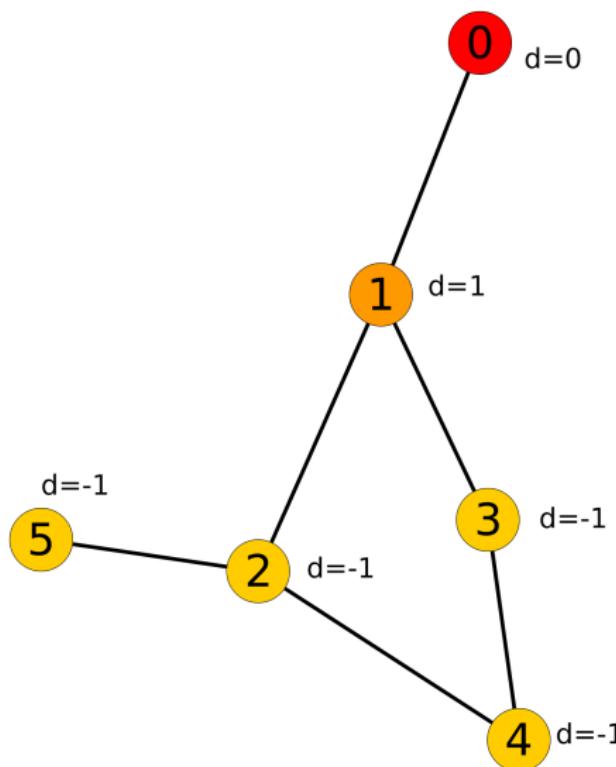


# INDIVIDUAZIONE DI UN CICLO (I)



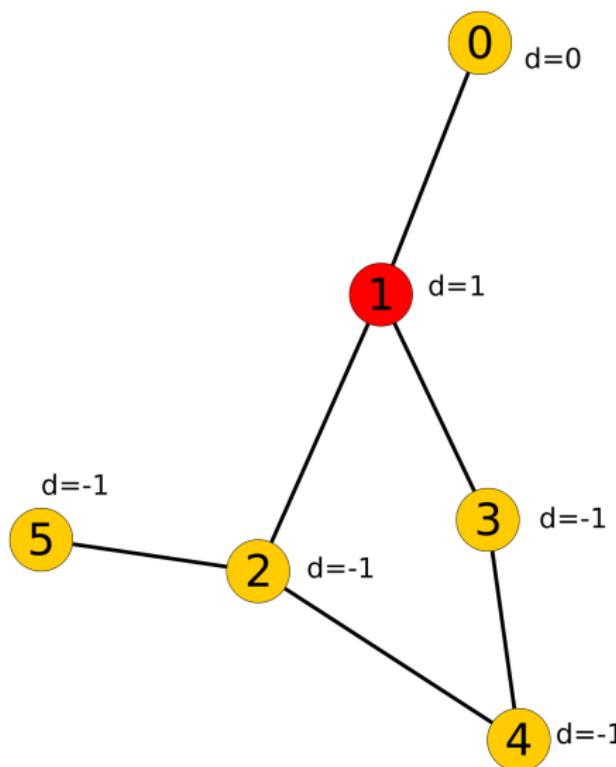
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)

# INDIVIDUAZIONE DI UN CICLO (I)



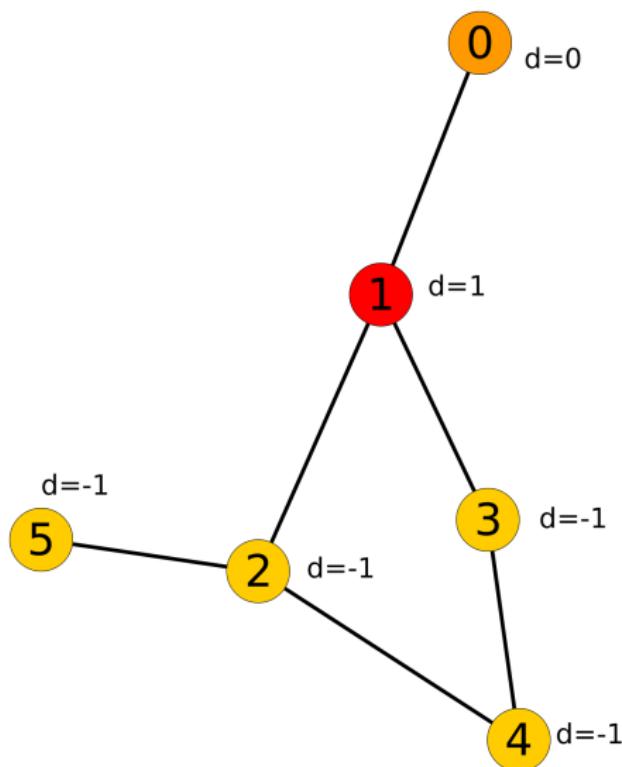
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)

## INDIVIDUAZIONE DI UN CICLO (II)



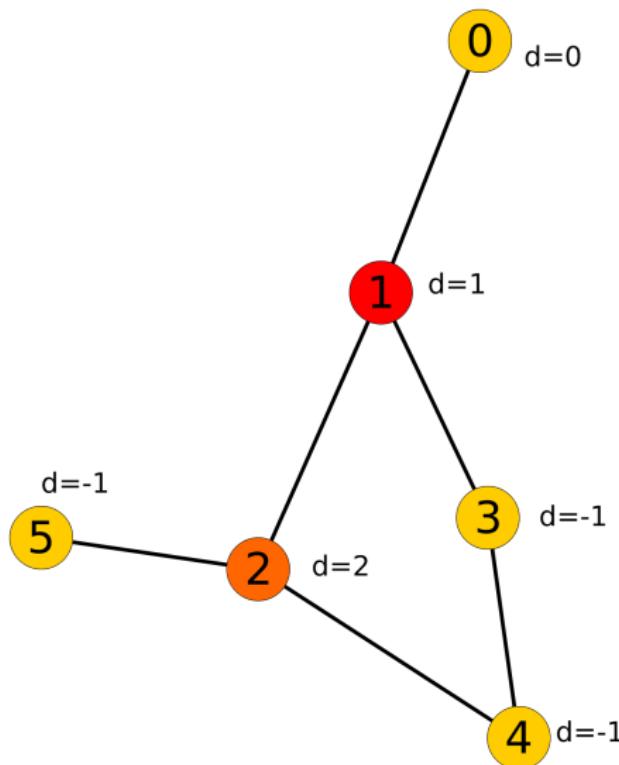
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1

# INDIVIDUAZIONE DI UN CICLO (II)



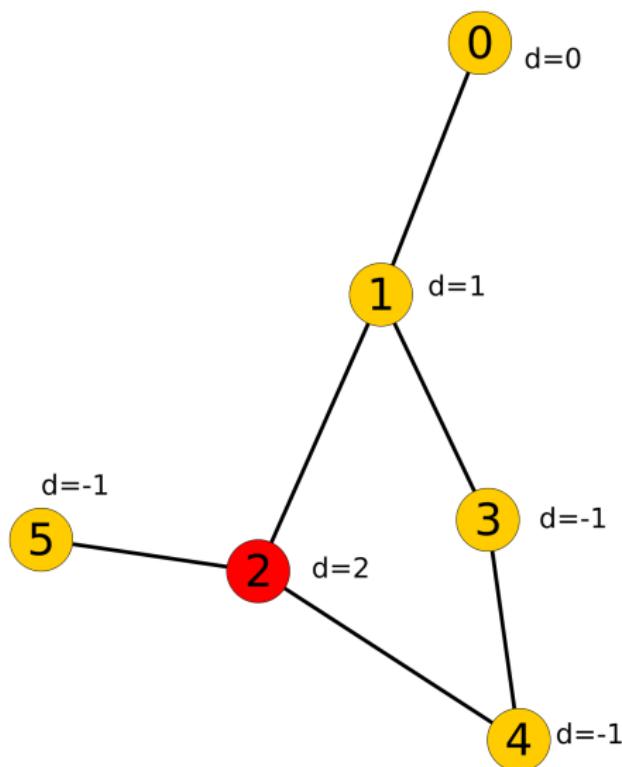
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1  $\rightarrow$  ignoro)

# INDIVIDUAZIONE DI UN CICLO (II)



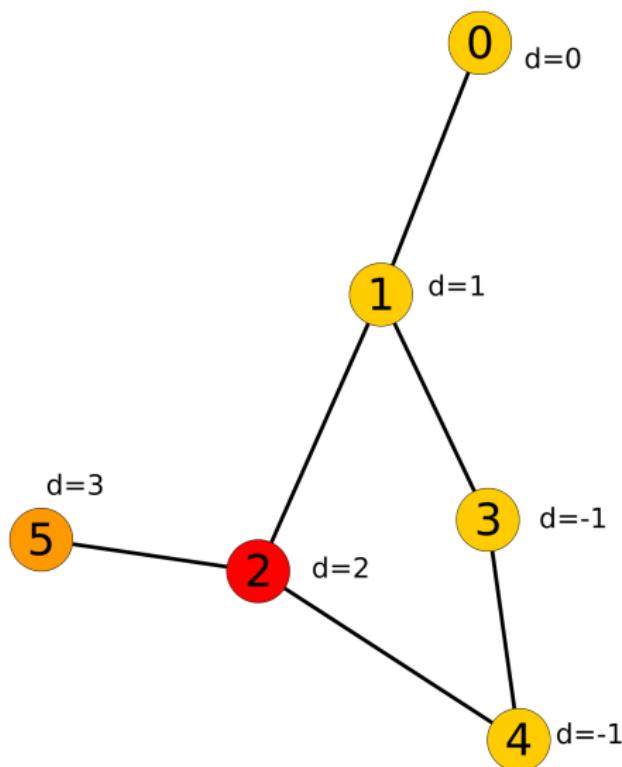
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 → ignoro)
- vicino 2, (profondità 2)

# INDIVIDUAZIONE DI UN CICLO (III)



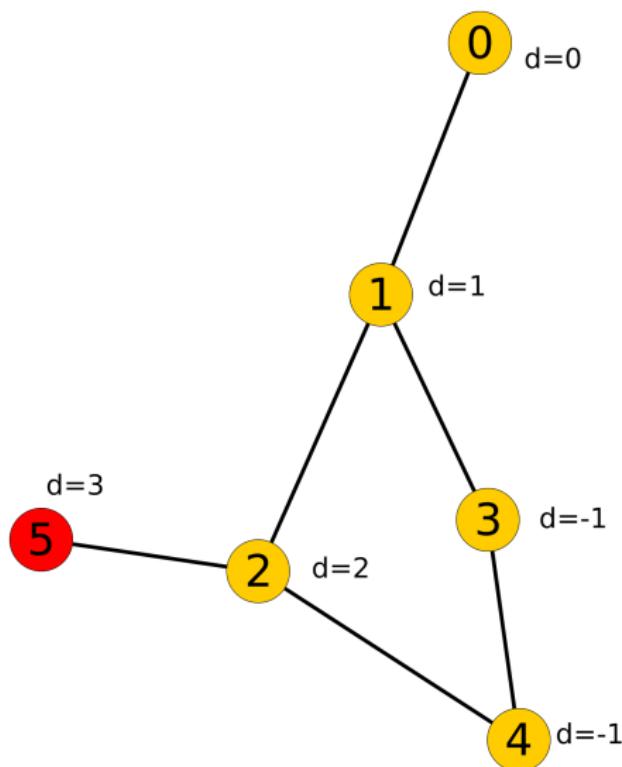
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 → ignoro)
- vicino 2, (profondità 2)
- visito 2

# INDIVIDUAZIONE DI UN CICLO (III)



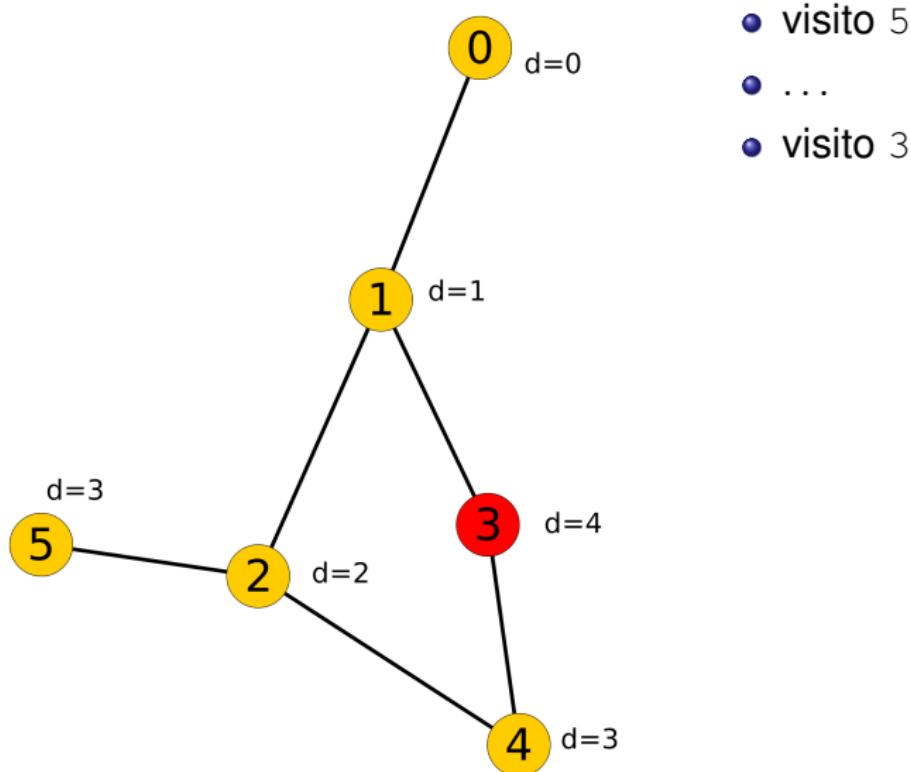
- nodi a profondità  $-1$
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 → ignoro)
- vicino 2, (profondità 2)
- visito 2
- vicino 5, (profondità 3)

# INDIVIDUAZIONE DI UN CICLO (IV)

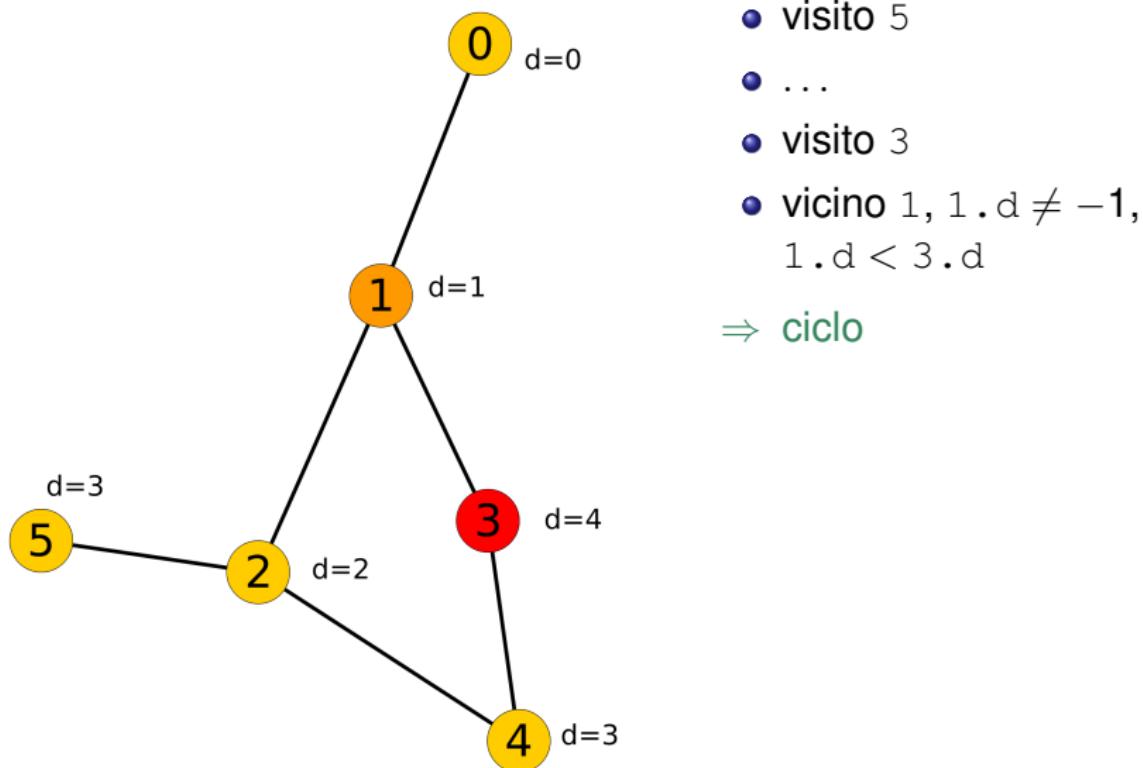


- nodi a profondità –1
- start, visito 0, (profondità 0)
- vicino 1 (profondità 1)
- visito 1
- vicino 0 (padre di 1 → ignoro)
- vicino 2, (profondità 2)
- visito 2
- vicino 5, (profondità 3)
- visito 5

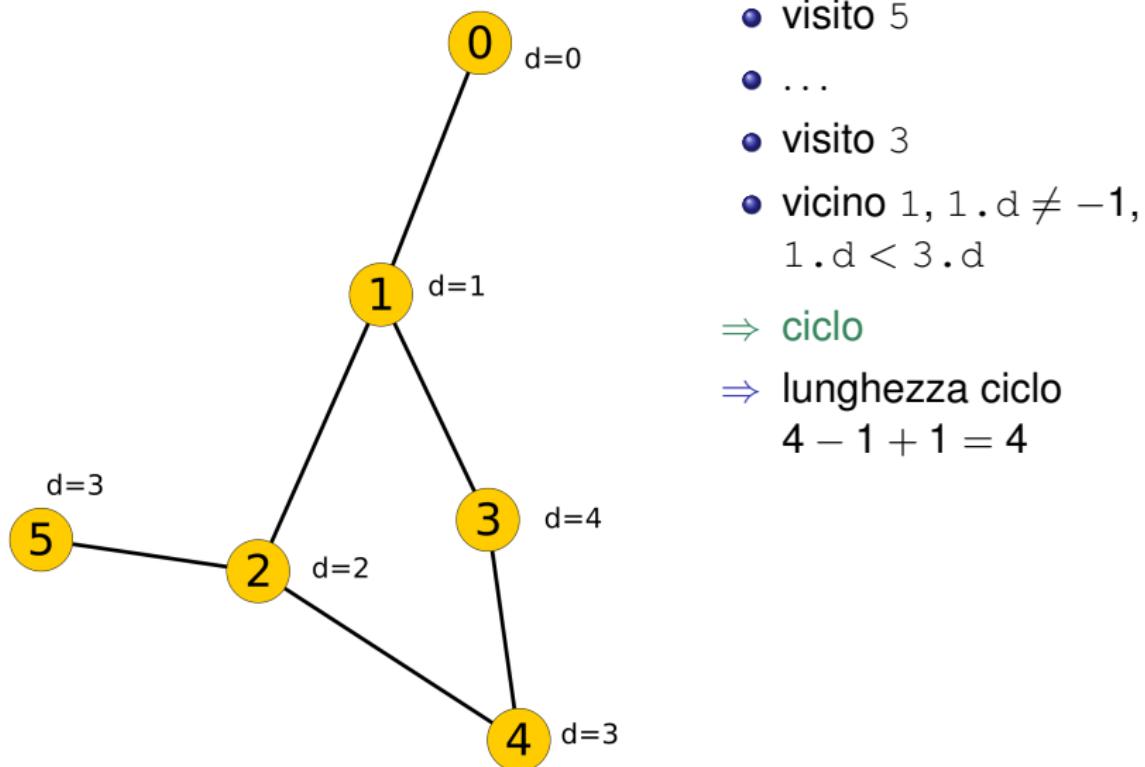
# INDIVIDUAZIONE DI UN CICLO (V)



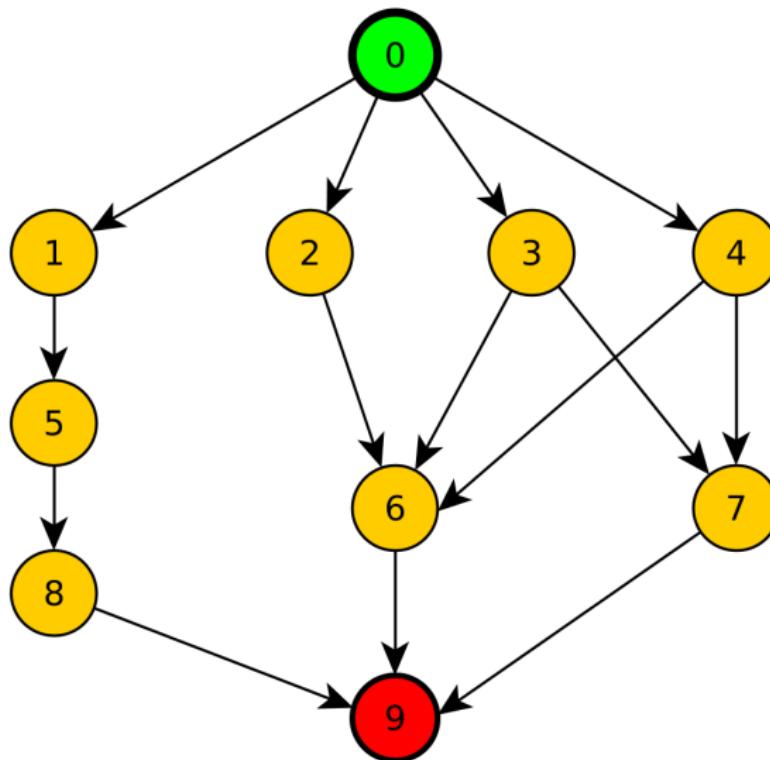
# INDIVIDUAZIONE DI UN CICLO (V)



# INDIVIDUAZIONE DI UN CICLO (VI)



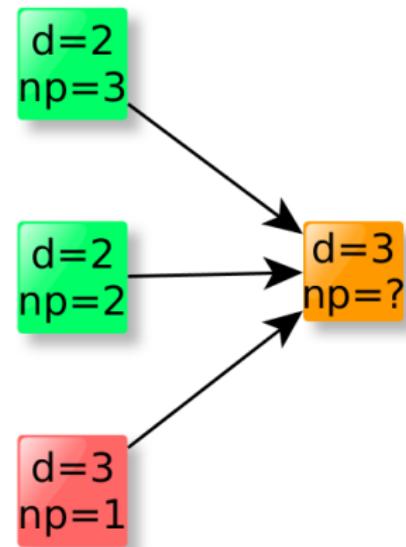
# SOLUZIONI: NUMERO DI CAMMINI MINIMI (NUMCAMMINI) (I)



# SOLUZIONI: NUMERO DI CAMMINI MINIMI (NUMCAMMINI) (II)

## IDEA

- Stiamo calcolando il numero di diversi cammini minimi da  $s$  a  $t$
- Sia  $P(v)$  l'insieme di predecessori di  $v$ , ovvero di tutti i  $w$  tali che:
  - ▶  $(w, v) \in E$
  - ▶  $d(s, v) = d(s, w) + 1$
- $NUMPATH(v) = \sum_{w \in P(v)} NUMPATH(w)$
- Calcoliamo  $NUMPATH(t)$



# ESERCIZI (I)

## DIMENSIONE MASSIMA COMPONENTE FORTEMENTE CONNESSA (COMPONENTE)

Dato un grafo orientato trovare la **dimensione** della massima componente fortemente connessa.

## ORDINAMENTO TOPOLOGICO (TOPORDER)

Dato un grafo diretto aciclico, trovare un suo ordinamento topologico.

## CAMMINO PIÙ LUNGO (CAMMINOLUNGO)

Dato un grafo diretto aciclico, trovare la lunghezza del suo cammino più lungo.

# ESERCIZI (II)

PRIMO PROGETTO A. A. 2021/2022

Super Mario Graph (`mario`)

PRIMO PROGETTO A. A. 2017/2018

007: Pioggia di laser (`laser`)