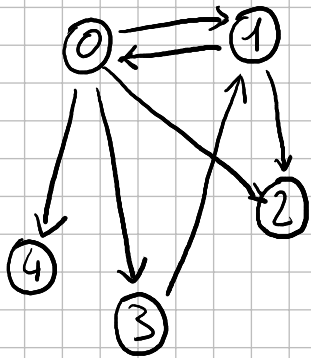
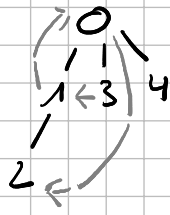


Archi negli alberi di visita

Grafo:

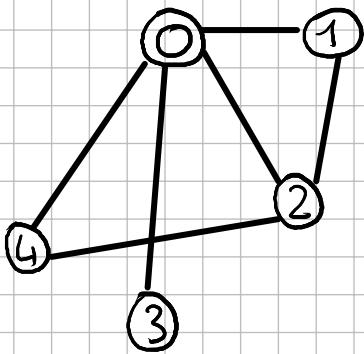


Albero di visita:

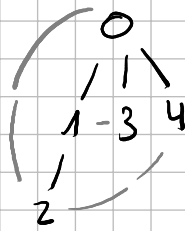


Gli archi in grigio non sono presenti nell'albero di visita, ma esistono nel grafo

Es su grafo non orientato

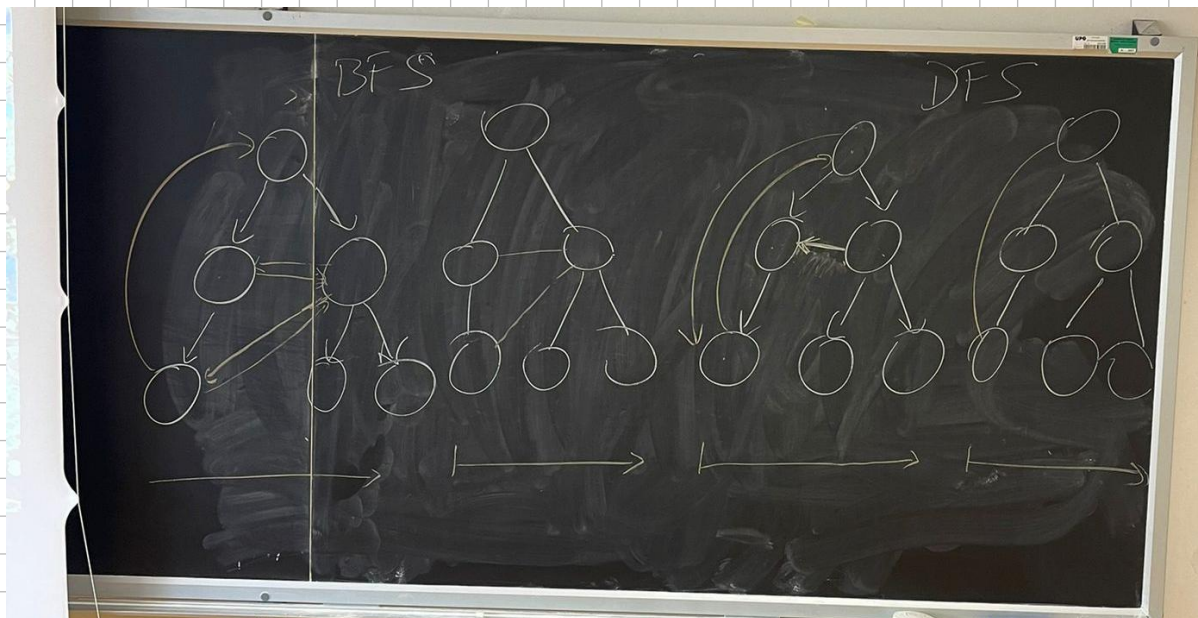


Albero



Esistono 3 tipi di archi

- Arco in avanti: (tra 0 e 2)
- Arco all'indietro: (tra 0 e 2)
- Arco trasversale: collega due sottoalberi distinti: (tra 1 e 3)



Visita DFS con nodi in ordine di fine visita

Visita DFS (s_{org})

$S = S \cup \{s_{org}\}$

per ogni vicino v di s_{org}

se $v \notin S$

$A = A \cup \{s_{org}, v\}$

visitaDFS(v)

$T = T \cup \{s_{org}\}$ // T tiene traccia dei nodi in ordine di fine visita

La visita termina sempre dopo che stati scoperti tutti i vicini di u ? Sì

Trovare i cicli all'interno di un grafo

VISITA DFS (s_{org})

```
{  
  S = S ∪ {sorg} ← INIZIO  
  PER OGNI VICINO v DI sorg  
    se v ∉ S  
      A = A ∪ {sorg, v}  
      visitaDFS(v)  
      OLTRE SE VICINO ∉ T  
      CICLO = TRUE  
  T = T ∪ {sorg} ← FINIS  
}
```

L'ORDINE TRA I NODI SCOPERTI

È SEMPRE TERMINANTI (DI FINI VISITA)
SONO DIVERSI NELLA DFS (NON BFS)

POSSIBILI APPLICAZ. TROVARE DEI CICLI
NEL GRAFO

Questo

implementazione

va bene solo

per grafi orientati:

perché in g.

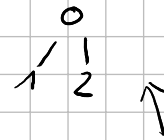
non orientati: trova
anche cicli banali!

Algoritmo per la ricerca di cicli (per grafi non orientati)

Per realizzare questo algoritmo dobbiamo capire cos'è l'array dei padri.

L'array dei padri ha una dimensione
pari al ordine del grafo.

Inizialmente è inizializzato a -1.



padre[0] = -1

padre[1] = 0

padre[2] = 0

Per ogni vicino di nodo

se vicino $\notin S$

padre[vicino] = nodo

// altre op della visita

altrimenti

se vicino \neq padre[nodo]

Ciclo = true

Posso farlo sia con BFS che con DFS

VISITA BFS (SORG)

$S = \{ \text{SORG} \}$

$\text{CODA} \leftarrow \text{SORG}$

FINCHÉ CODA NON VUOTA

$\text{NODO} \leftarrow \text{CODA}$

PER OGNI VICINO DI NODO

SE VICINO $\notin S$

$S = S \cup \{\text{vicino}\}$

$\text{CODA} \leftarrow \text{VICINO}$

$\text{PADRE}[\text{VICINO}] = \text{NODO}$

CAMMINO Min(A,B)

VISITA BFS(A)

$\text{TEMP} = B$

WHILE $\text{PADRE}[\text{TEMP}] \neq -1$

$\text{CAMMINO} \leftarrow \text{ADD}(\text{TEMP})$

$\text{TEMP} = \text{PADRE}[\text{TEMP}]$

$\text{CAMMINO} \leftarrow \text{ADD}(\text{TEMP})$

POI VA INVERTITO (ANCHE
ANCHE ORA FINO)



B C D A

A D C B