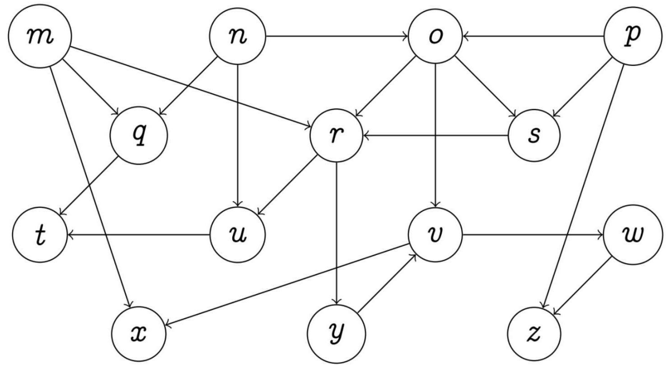
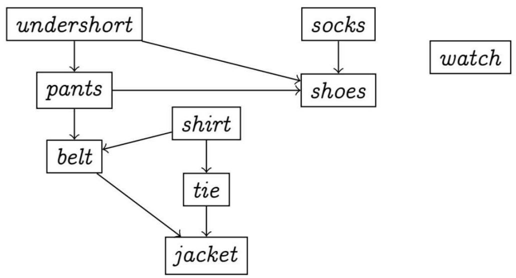
Grafo orientato aciclico (directed acyclic graph: DAG)



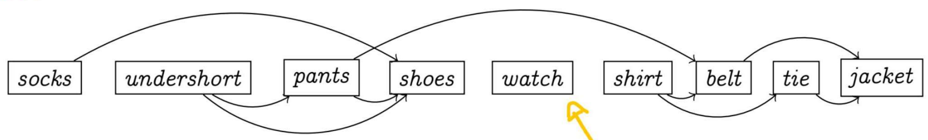
Ordinamento topologico: definizione

- ▷ Sia $G = (V, E)$ un grafo orientato e aciclico (non pesato)
- ▷ Ordinare i nodi di G in modo tale che se $(u, v) \in E$ allora u compare prima di v nell'ordinamento

Ordinamento topologico (topological sort): esempio



STESURA MONODIMENSIONALE CON ARCHI SOLO IN AVANTI



IMPOSSIBILE CON CICLI

POSIZIONE QUALSIASI

Ordinamento topologico

TOPOLOGICAL-SORT(G)

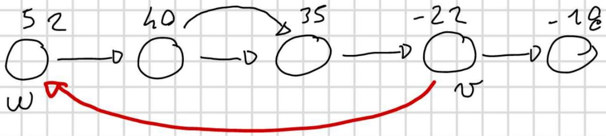
```
call DFS(G) to compute finishing times v.f for each vertex v
as each vertex is finished, insert it onto the front of a linked list
return the linked list of vertices
```

$\Theta(n+m)$

ORDINAMENTO DECRESCENTE FINE VISITA

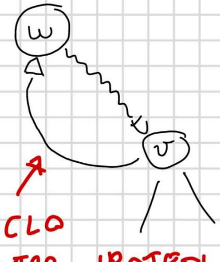
DIM PER ASSURDO

Supponiamo rimanga un arco all'indietro $(v,w) \in E$

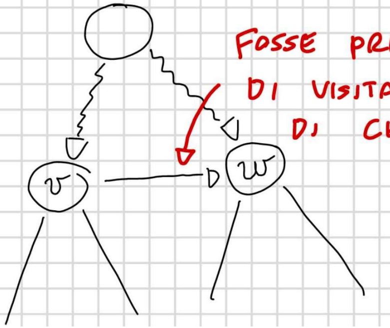


Ma i casi in cui può succedere sono

w padre

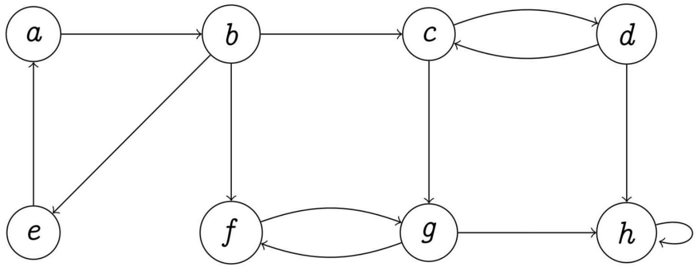


CICLO CONTRO IPOTESI



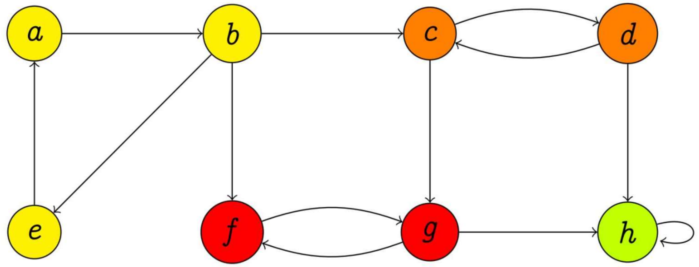
FOSSE PRESENTE FINIREI DI VISITARE w PRIMA DI CHIUDERE v

Grafo di partenza



Componenti fortemente connesse (strongly connected components)

↳ DA OGNI NODO POSSO ARRIVARE AD UN ALTRO CON UN CAMMINO

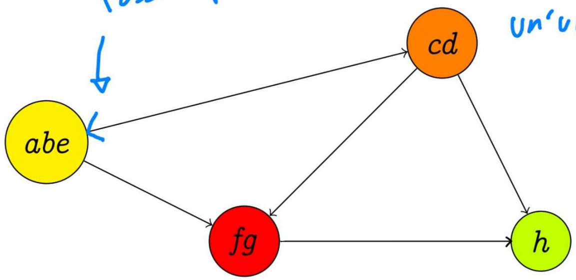


OBBIETTIVO: Trovare famiglie di componenti fortemente connessi

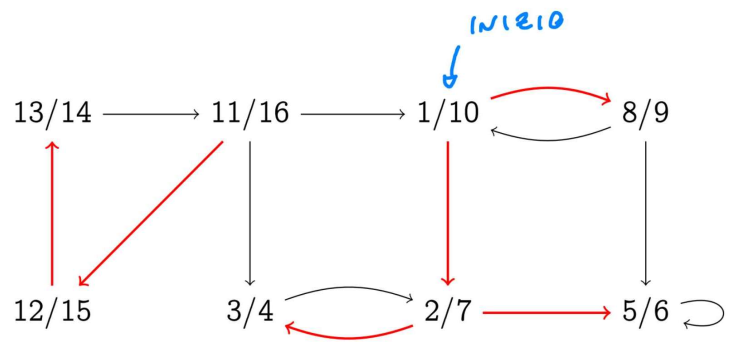
Componenti fortemente connesse

ACICLICO

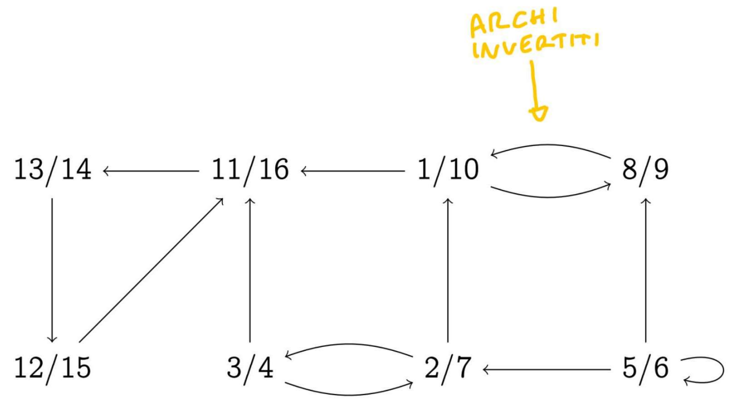
Fosse presente abecd sarebbe un'unica famiglia



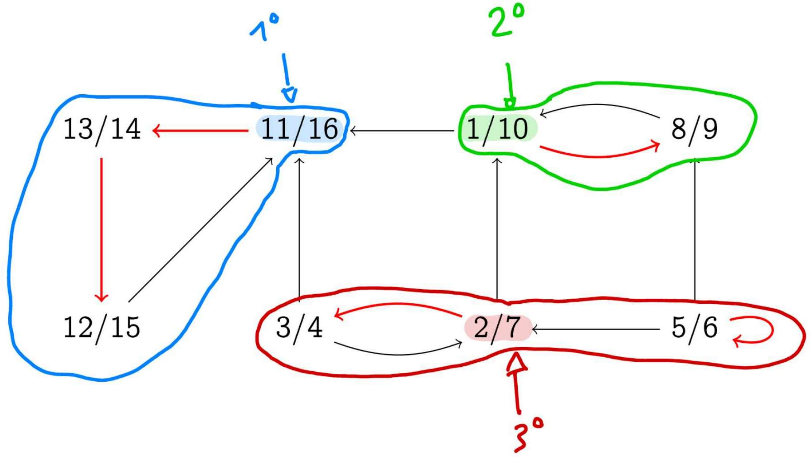
DFS: tempi di inizio e fine visita



Grafo trasposto MATRICE ADIACENZA TRASPOSTA



DFS del grafo trasposto partendo dai nodi in ordine decrescente di fine visita: componenti connesse



Insiemi di nodi massimali: definizione

- Sia $G = (V, E)$ un grafo orientato (non pesato)
- Un insieme $S \subseteq V$ è massimale rispetto a una data proprietà P se la proprietà vale per S ma non vale più se aggiungiamo qualsiasi altro nodo a S
- Sia P la seguente proprietà di $S \subseteq V$: per ogni coppia di nodi (u, v) appartenenti a S esiste una coppia di cammini $u \rightsquigarrow v$ e $v \rightsquigarrow u$ in G.

Problema delle componenti fortemente connesse: definizione

- ▷ Input: un grafo orientato $G = (V, E)$
- ▷ Output: una partizione di V in insiemi massimali rispetto alla proprietà P (componenti fortemente connesse)
- ▷ Si noti che la soluzione è unica

Componenti fortemente connesse

```
STRONGLY-CONNECTED-COMPONENTS(G)
1  DFS(G) // compute for each node u its finishing time u.end
2  G^T = TRANSPOSE(G)   Θ(m)
3  SCC-DFS(G^T)
4  return components computed by SCC-DFS at step 3
```

$\Theta(n+m)$

DFS modificata

```
SCC-DFS(G)
for each vertex u ∈ G.V
    u.visited = FALSE
    u.π = NIL
time = 0
for each vertex u ∈ G.V sorted by decreasing finishing times
    if u.visited == FALSE
        DFS-SINGLE-COMPONENT(G, u)
```

ORDINATI TRAMITE DFS

$\Theta(n+m)$