VISITA DFS CON CLOCK

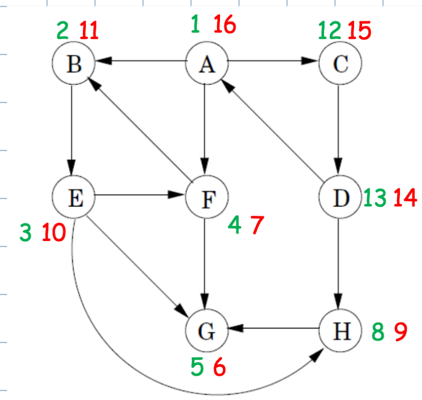
UTILIZZIAMO 2 ATTRIBUTI CONTENUTI NEL NODO:

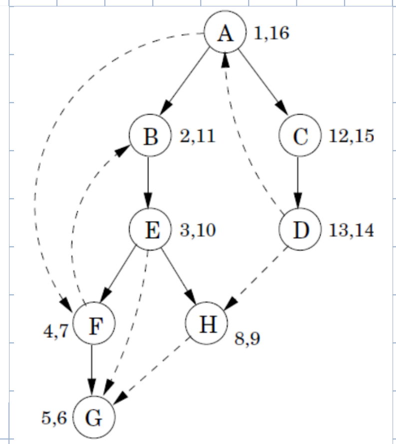
- PRE(v) ⟶ INDICA A CHE ISTANTE DI CLOCK ABBIAMO <u>VISITATO</u> v
- POST(v) ⟶ INDICA A CHE ISTANTE STIAMO <u>USCENDO</u> DA v

- CLOCK ⟶ VALORE INTERO ≥ 1

procedura visitaDFSRicorsiva(*vertice v, albero T*)
  marca e visita il vertice v ← pre(v)=clock, clock=clock+1
  for each (arco (v, w)) **do**
    if (w non è marcato) **then**
      aggiungi l'arco (v, w) all'albero T
      visitaDFSRicorsiva(w, T)
  post(v)=clock; clock=clock+1

algoritmo visitaDFS(*vertice s*) → *albero*
  T ← albero vuoto ← clock=1
  visitaDFSRicorsiva(s, T)
  return T





DATI 2 NODI U e V, ABBIAMO CHE U È ANTENATO DI V SE:

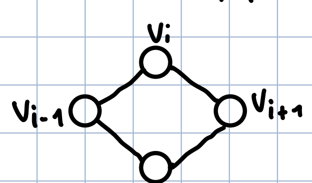
- PRE(U) < PRE(V) ⟶ SIGNIFICA CHE ABBIAMO VISITATO PRIMA <u>U</u> E POI <u>V</u>
- POST(U) > POST(V) ⟶ SIGNIFICA CHE SIAMO USCITI PRIMA DA <u>V</u> E POI DA <u>U</u>

VALE QUINDI: PRE(U) < PRE(V) < POST(V) < POST(U)

CICLI NEI GRAFI

UN GRAFO G HA UN CICLO SE E SOLO SE LA VISITA DFS RIVELA UN ARCO ALL'INDIETRO.

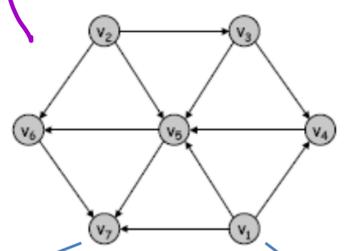
SE C'È UN CICLO CI SARÀ UN CAMMINO $\langle v_0, v_1, \dots, v_k = v_0 \rangle$.
SIA v_i IL PRIMO NODO VISITATO, POICHÉ v_{i-1} È RAGGIUNGIBILE DA v_i
ALLORA (v_{i-1}, v_i) È UN ARCO ALL'INDIETRO.



GRAFO DIRETTO ACICLICO (DAG) → GRAFO DIRETTO CHE NON HA CICLI.

↳ ORDINAMENTO TOPOLOGICO → DISPOSIZIONE LINEARE DI NODI IN UN DAG TALE CHE PER OGNI ARCO (U,V) IL NODO U PRECEDE V.

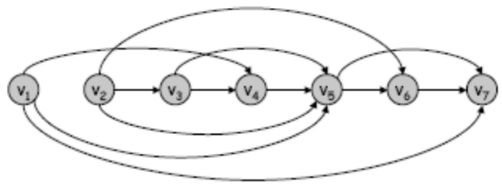
DAG



pozzo: solo archi entranti

a DAG

sorgente: solo archi uscenti



a topological ordering

ORDINAMENTO TOPOLOGICO

UN GRAFO G AMMETTE UN ORDINAMENTO TOPOLOGICO ⇔ È UN DAG.
CI SONO DUE MODI CON I QUALI POSSIAMO EFFETTUARE UN ORDINAMENTO TOPOLOGICO.

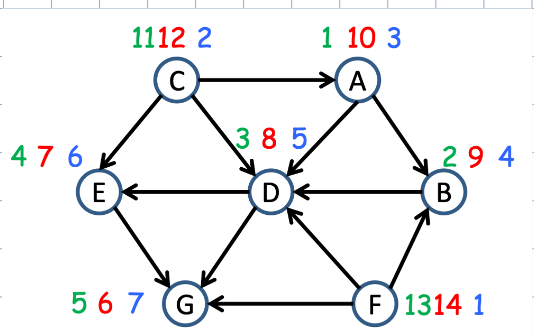
1 METODO (DFS)

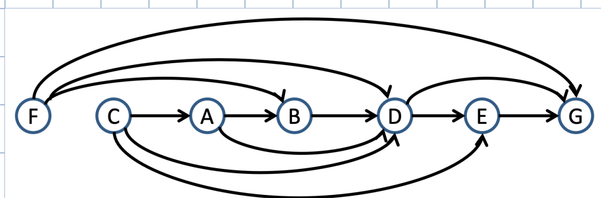
EFFETTUIAMO UNA RICERCA DFS E RESTITUIAMO IN ORDINE DECRESCENTE IN BASE AL TEMPO DI FINE VISITA POST.

OrdinamentoTopologico (grafo *G*)

1. top=n; *L* ← lista vuota;
2. chiama visita DFS ma:
   1. quando hai finito di visitare un nodo *v* (quando imposti post(v)):
   2. σ(v)=top; top=top-1;
   3. aggiungi v in testa alla lista L
3. **return** *L* e σ

SAPPIAMO CHE DATO UN ARCO (U, V) DIRETTO, IL POST (U) > POST (V), AVENDO POI CHE Ⓤ→Ⓥ QUINDI V HA UNA "DIPENDENZA" DA U, ORDINANDO IN MODO DECRESCENTE PER POST, I NODI CHE SARANNO NECESSARI PER ALTRI NODI VERRANNO MESSI PRIMA.
↳ (USCIRANNO DOPO DALLA DFS)





COMPLESSITÀ: $\Theta(n+m)$ CON LISTE DI ADIACENZA

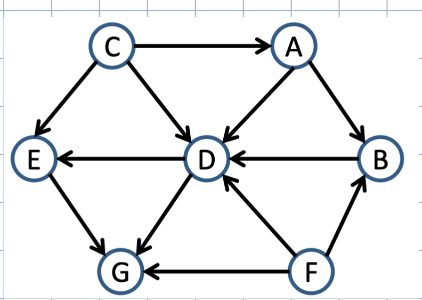
NOTA: NON POSSIAMO AVERE ARCHI ALL' INDIETRO, ANCHE XCHÉ È UN DAG.

2 METODO (ALTERNATIVO)

CLONIAMO IL GRAFO IN UN NUOVO GRAFO G', FINCHÉ CI SONO NODI SENZA ARCHI ENTRANTI, RIMUOVIAMO L'I-ESIMO NODO E LO AGGIUNGIAMO IN FONDO ALLA LISTA, DOPODICHÉ RIMUOVIAMO QUEL NODO CON I RELATIVI ARCHI USCENTI.
COSÍ FACENDO MANO MANO CHE RIMUOVIAMO I NODI SIGNIFICA CHE STIAMO RIMUOVENDO DA G' E METTENDO NELLA LISTA, UN NODO CHE NON HA DIPENDENZE.
INIZIAMO DAI NODI SORGENTI E MANO MANO LIBERIAMO LE DIPENDENZE DEGLI ALTRI NODI.
SE G' AL TERMINE DEL CICLO NON È PIÚ VUOTO ALLORA SIGNIFICA CHE G NON È ACICLICO.

algoritmo ordinamentoTopologico($grafo\ G$) $\rightarrow$ $lista$
    $\widehat{G} \leftarrow G$
    $ord \leftarrow$ lista vuota di vertici
    while (esiste un vertice u senza archi entranti in $\widehat{G}$) **do**
        appendi u come ultimo elemento di ord
        rimuovi da $\widehat{G}$ il vertice u e tutti i suoi archi uscenti
(*) **if** ($\widehat{G}$ non è diventato vuoto) **then errore** il grafo G non è aciclico
    return ord

COMPLESSITÀ: $\Theta(n+m)$ CON LISTE DI ADIACENZA



- TOLGO F   L = [F]
- TOLGO C   L = [F, C]
- TOLGO A   L = [F, C, A]
- TOLGO B   L = [F, C, A, B]
- TOLGO D   L = [F, C, A, B, D]
- TOLGO E   L = [F, C, A, B, D, E]
- TOLGO G   L = [F, C, A, B, D, E, G]

ORDINE TOPOLOGICO: F, C, A, B, D, E, G

COMPONENTI FORTEMENTE CONNESSE

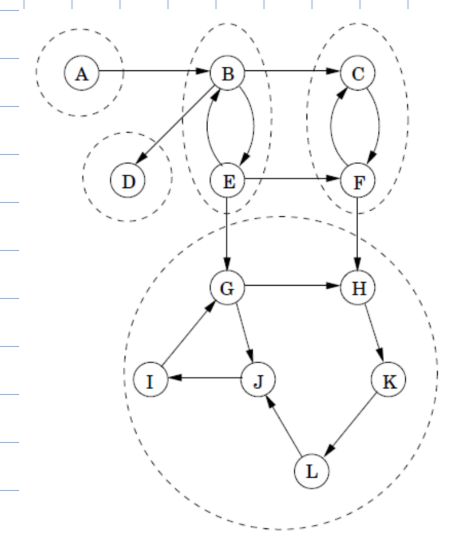
DUE VERTICI U e V SONO FORTEMENTE CONNESSI SE:

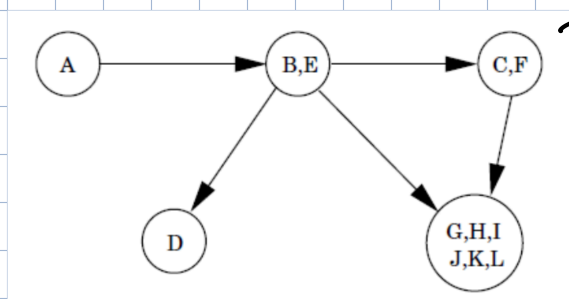
- ESISTE UN CAMMINO DA U a V
- ESISTE UN CAMMINO DA V a U

} ATTENZIONE, UN CAMMINO, NON UN ARCO

UNA COMPONENTE FORTEMENTE CONNESSA È UN INSIEME $C \subseteq V$ T.C. OGNI COPPIA DI NODI SONO FORTEMENTE CONNESSI.

↳ QUESTO INSIEME È MASSIMALE CIOÈ SE AGGIUNGIAMO UN ALTRO NODO A C LA PROPRIETÀ NON È PIÙ VERIFICATA.





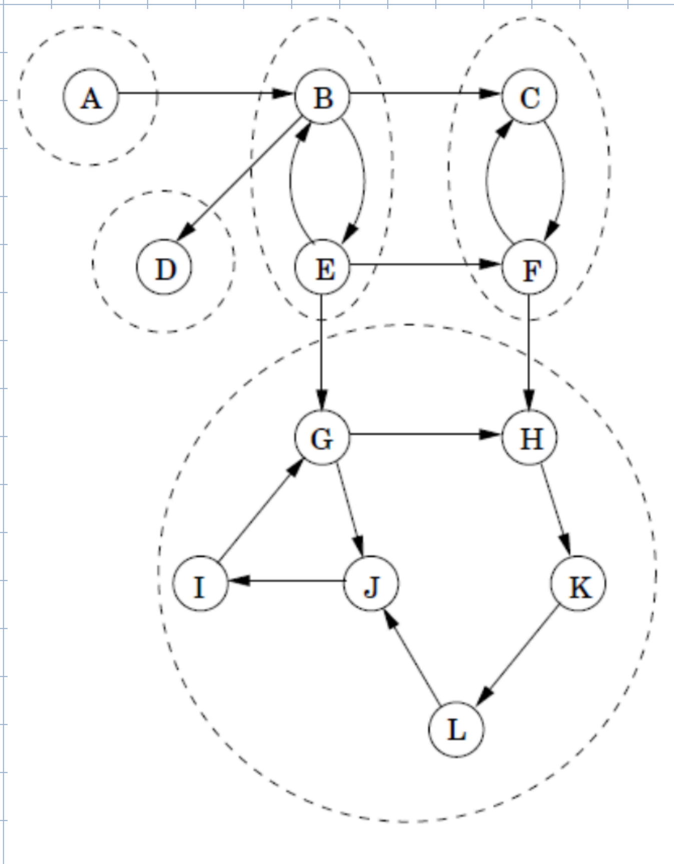
↷ DATI DUE NODI C e C', C'È UN ARCO DA C a C' SE C'È UN ARCO DA UN NODO IN C VERSO UN NODO IN C'.

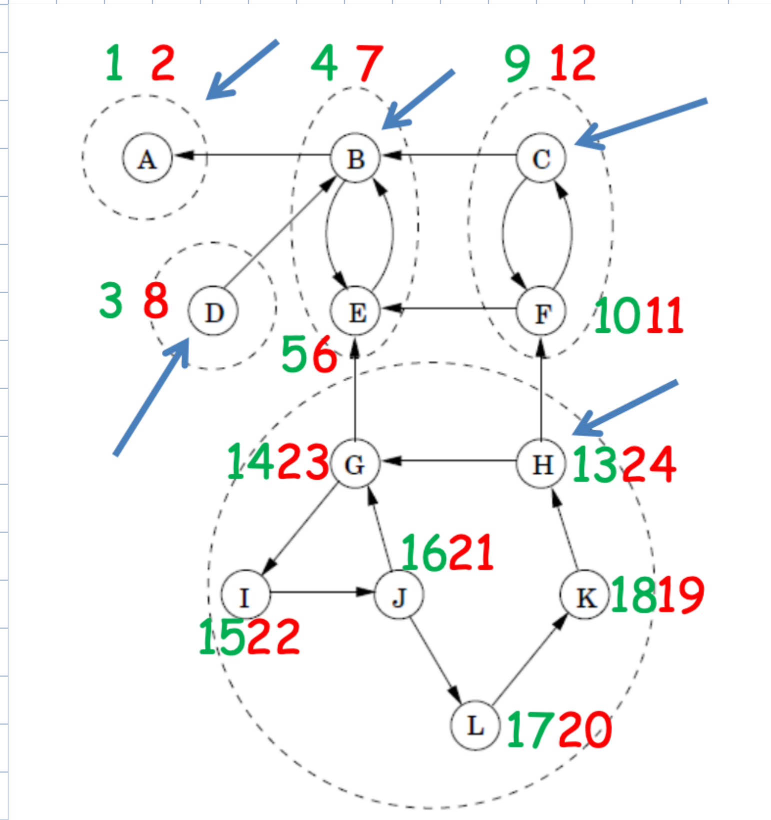
GRAFO COMPONENTI FORT. CONN.

↓

È SEMPRE UN DAG

CALCOLARE COMPONENTI IN UN GRAFO DIRETTO





H, G, I, J, L, K, C, F, D, B, E, A

Comp:

H, K, L, J, I, G, J

C, F

D

B, E

A