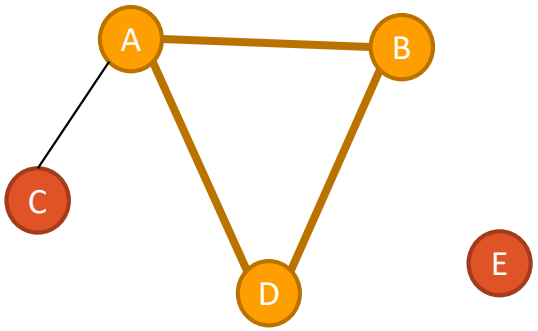
GRAFI: TEST DI ACICLICITÀ

Questa parte non è presente in nessuno dei testi adottati.

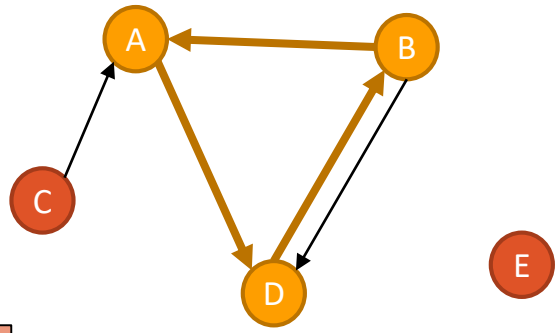
(dall'introduzione ai grafi) Ciclo

Un cammino $\langle w_1, w_2, \dots, w_n \rangle$ si dice **chiuso** se $w_1 = w_n$. Un cammino chiuso, semplice, di lunghezza almeno 1 si dice **ciclo**.

$\langle A, B, D, A \rangle$ è un ciclo



$\langle A, D, B, A \rangle$ è un ciclo

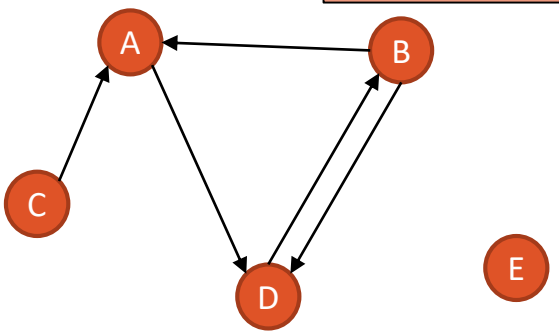


NUOVA DEFINIZIONE.
Cappio: un cappio è un arco che collega un vertice a se stesso.

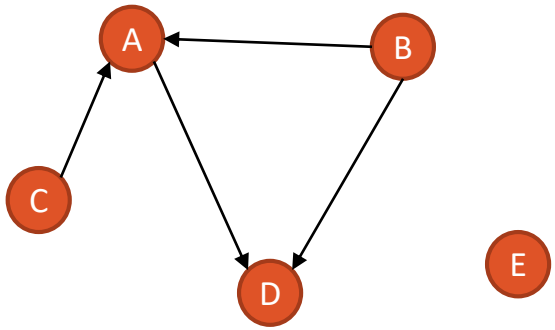
(dall'introduzione ai grafi) Grafo aciclico

Se un grafo non contiene cicli, si dice **aciclico**

Questo grafo orientato contiene il ciclo $\langle A, D, B, A \rangle$ (quindi non è aciclico)



Rimuovendo l'arco $\langle D, B \rangle$ il grafo diventa aciclico



Generalità delle visite

Le visite che abbiamo considerato fino ad ora possono essere applicate (senza modifiche particolari) a grafi orientati e non orientati.

Ora vediamo una possibile applicazione della visita DFS per verificare **l'aciclicità di un grafo**.

In particolare, usiamo una **visita DFS** per **classificare** gli archi di un grafo, se troviamo determinati tipi di archi il grafo è **ciclico**.

In questo caso, l'algoritmo per grafi orientati e quello per grafi non orientati differiscono (di poco), prestate attenzione al tipo di grafo di cui stiamo parlando.

NB: per generalità, consideriamo grafi **non connessi**.

Classificazione degli archi di un grafo **orientato** durante una DFS

Diciamo che un arco ⟨u, v⟩ viene "**percorso**" quando si incontra v nella lista degli adiacenti ad u. Si noti che durante una DFS di un grafo orientato ogni arco è percorso esattamente una volta.

DEFINIAMO

Arco dell'albero: arco **inserito nella foresta DFS** (quando è percorso, scoprendo il vertice di arrivo)

Arco all'indietro: arco che, collega **un vertice ad un suo antenato** in un albero della foresta DFS (i cappi sono considerati archi all'indietro)

Arco in avanti: arco che collega **un vertice ad un suo discendente** in un albero della foresta DFS

Arco di attraversamento: arco che collega due vertici che non sono in relazione antenato - discendente

Criterio di Classificazione con visita DFS (grafi **orientati**)

Durante la visita di un grafo orientato un arco ⟨u, v⟩ viene "percorso" quando si incontra v nella lista degli adiacenti ad u. In quel momento color[v] può essere:

bianco: ⟨u, v⟩ è un **arco dell'albero**

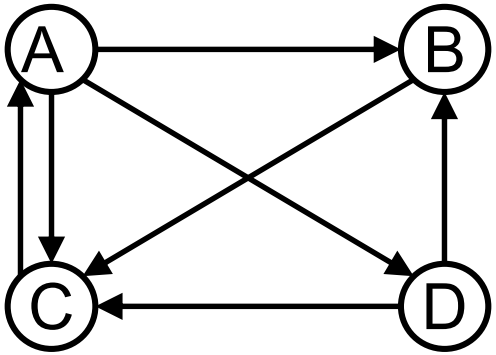
grigio: u è un discendente di v in un albero della foresta DFS, ⟨u, v⟩ è un **arco all'indietro**

nero: la visita di v è già terminata, ⟨u, v⟩ è un arco:

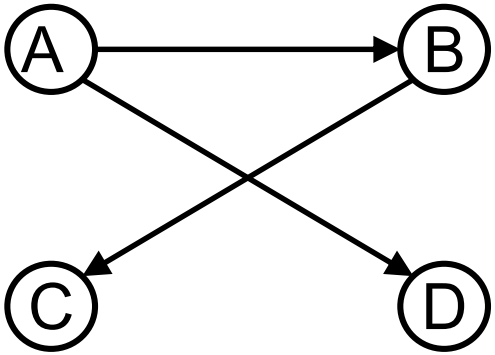
- **in avanti** se v è un discendente di u, in tal caso d[u] < d[v] < f[v] < f[u], ovvero: d[u] < d[v]
- **di attraversamento** altrimenti, in tal caso d[v] < f[v] < d[u] < f[u], ovvero: d[v] < d[u]

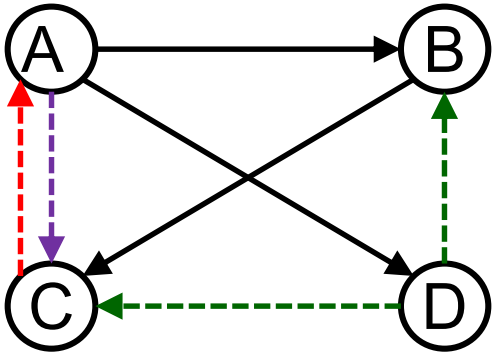
Esempio

grafo



albero di visita in profondità da A





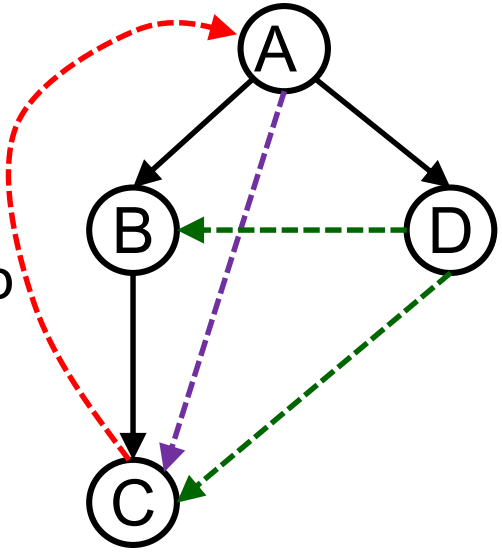
AC: in avanti

CA: all'indietro

DB: di attraversamento

DC: di attraversamento

Tutti gli altri: dell'albero



Classificazione degli archi di un grafo **non orientato** durante una DFS

Diciamo che un arco (u, v) viene "**percorso**" quando si incontra v nella lista degli adiacenti ad u. Si noti che durante una DFS di un grafo orientato ogni arco è percorso esattamente due volte. Di seguito classifichiamo archi percorsi per **la prima volta**.

DEFINIAMO

Arco dell'albero: arco inserito nella foresta DFS (quando è percorso per la prima volta, scoprendo il vertice di arrivo)

Arco all'indietro: arco che collega un vertice ad un suo antenato in un albero della foresta DFS (i cappi sono considerati archi all'indietro)

NOTA: non possono esistere archi in avanti o di attraversamento.

Criterio di Classificazione con visita DFS (grafi **non orientati**)

Durante la visita di un grafo non orientato un arco (u, v) viene "percorso" quando si incontra v nella lista degli adiacenti ad u. In quel momento color[v] può essere:

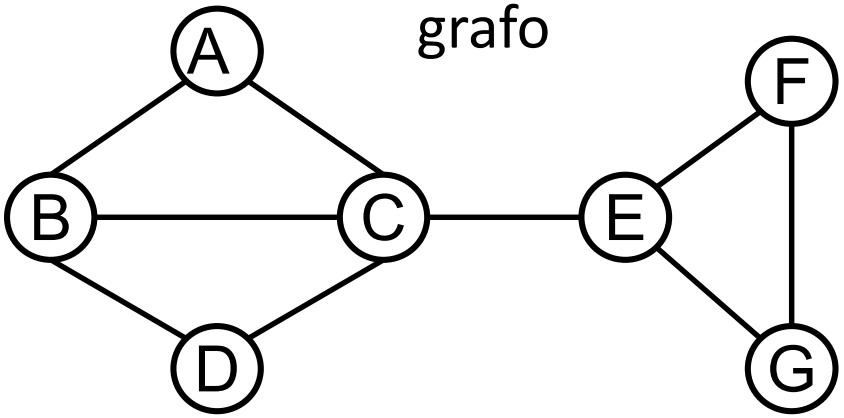
bianco: (u, v) non è mai stato percorso e classificato. v viene inserito nell'albero come figlio di u e (u, v) è un **arco dell'albero.**

grigio: v è ancora aperto quindi u è un discendente di v in un albero della foresta DFS. Di conseguenza, (u, v) è un **arco all'indietro.**

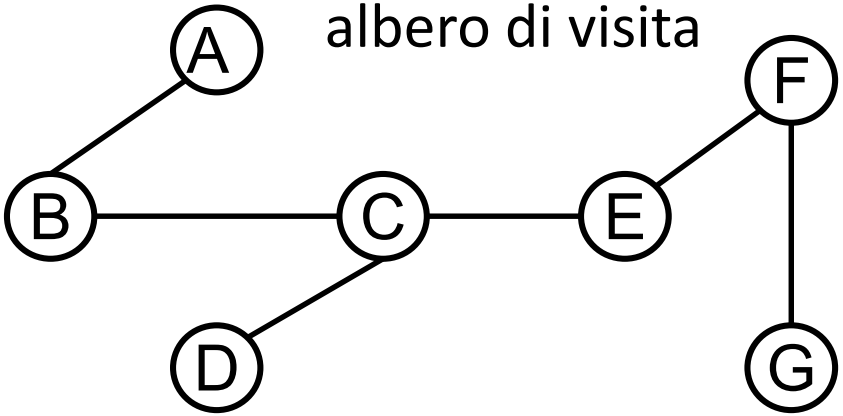
nero: la visita di v è già terminata, (u, v) è già stato percorso (nella direzione opposta) e quindi è già stato classificato (come **arco all'indietro**).

Esempio

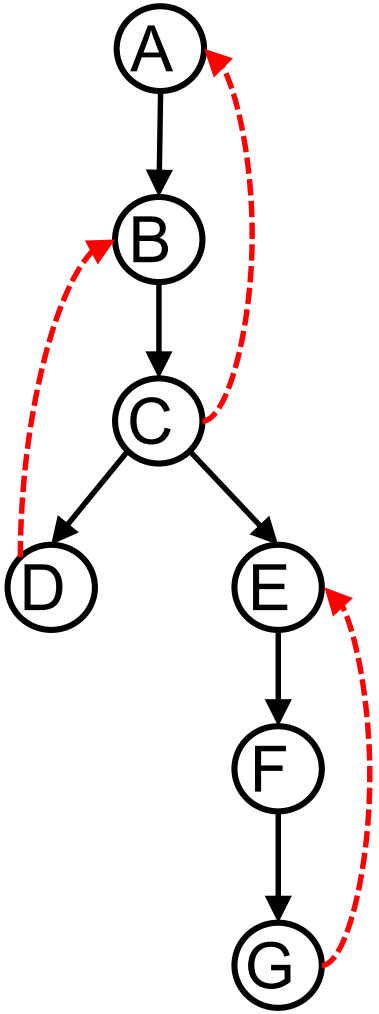
grafo



albero di visita



CA, DB, GE: archi all'indietro



Teorema del grafo aciclico

Se un grafo (orientato o non orientato) G **contiene un ciclo**, rispetto a una qualunque visita in profondità di G esiste un **arco all'indietro**.

Viceversa, se una visita in profondità di un grafo (orientato o non orientato) G produce un **arco all'indietro**, il grafo G **contiene un ciclo**.

Quindi:

Un grafo, orientato o non orientato , è **aciclico** se e solo se una visita DFS (qualunque) **non produce archi all'indietro**.

Come si implementa nella pratica?

(A)ciclicità nei grafi non orientati

```
boolean CICLICO(G)
   INIZIALIZZA(G)
   for ogni nodo u di G
      if color[u] = white and VISITA-ric-ciclo(G, u)
         return true
   return false

boolean VISITA-ric-ciclo(G, u)
   color [u] <- gray
   for ogni v adiacente ad u
      if color[v] = white
         π[v] <- u
         if VISITA-ric-ciclo (G, v) return true
      else if (v ≠ π[u]) return true
   color[u] <- black
   return false
```

Se la visita DFS trova un vertice v visitato che non sia il padre di u (perché in questo caso (u,v) è già stato classificato percorrendolo nell'altro senso), è stato rilevato un ciclo e l'algoritmo restituisce true

(A)ciclicità nei grafi orientati

```
boolean CICLICO(G)
   INIZIALIZZA(G)
   for ogni nodo u di G
      if color[u] = white and VISITA-ric-ciclo(G, u)
         return true
   return false

boolean VISITA-ric-ciclo(G, u)
   color [u] <- gray
   for ogni v adiacente ad u
      if color[v] = white
         π[v] <- u
         if VISITA-ric-ciclo (G, v) return true
      else if color[v] = gray return true
   color[u] <- black
   return false
```

Ricorda: $\langle u, v \rangle \neq \langle v, u \rangle$ quindi non bisogna controllare che v non sia il padre di u

Se v è nero, $\langle u, v \rangle$ è un arco di attraversamento o in avanti, quindi non rappresenta un ciclo

Complessità del test di aciclicità

Poiché si basa su una visita DFS, la complessità del test di aciclicità è **O(n+m)**.

Cosa devo aver capito fino ad ora

- Come utilizzare una visita DFS per classificare gli archi di
  - Un grafo orientato
  - Un grafo non orientato
- Archi dell'albero, archi in avanti, archi all'indietro, archi di attraversamento – come riconoscerli e loro «significato»
- Come usare la classificazione per implementare un test di aciclicità per grafi orientati e non orientati

…se non ho capito qualcosa

- Alzo la mano e chiedo
- Ripasso sul libro
- Chiedo aiuto sul forum
- Chiedo o mando una mail al docente