Componenti fortemente Connesse

Nei grafi DIRETTI, il concetto di componente connessa non è molto significativo, poiché

$u \rightsquigarrow v$ non implica che $v \rightsquigarrow u$

quindi la nozione di connessione non è una RELAZIONE di EQUIVALENZA

Def Componente fortemente connessa di un grafo DIRETTO

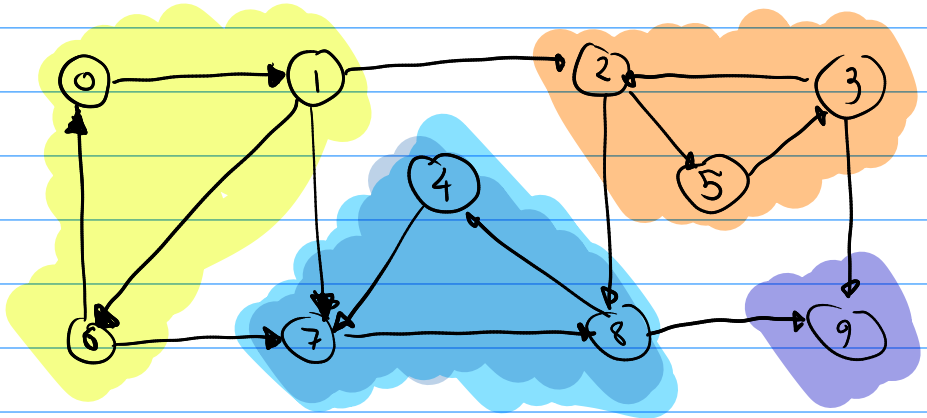
- Un insieme MASSIMALE $U \subseteq V(G)$
- per ogni $u, v \in U$ si ha che $u \rightsquigarrow v$ e $v \rightsquigarrow u$

dove $w_1 \rightsquigarrow w_2$ indica che esiste un cammino da w_1 a w_2

alternativamente definiamo $u \sim v$ quando $u \rightsquigarrow v$ e $v \rightsquigarrow u$

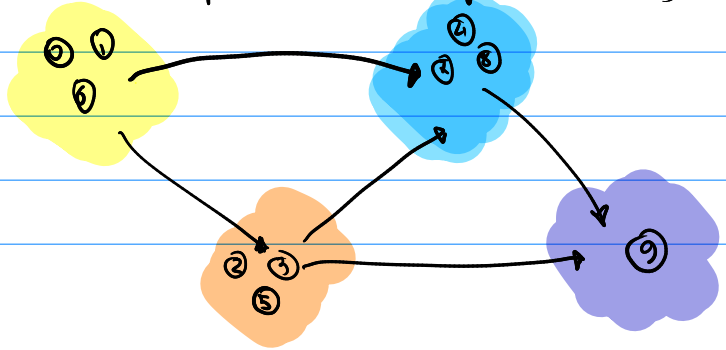
allora le componenti connesse di G sono le classi di equivalenza di $\sim$

E.g.



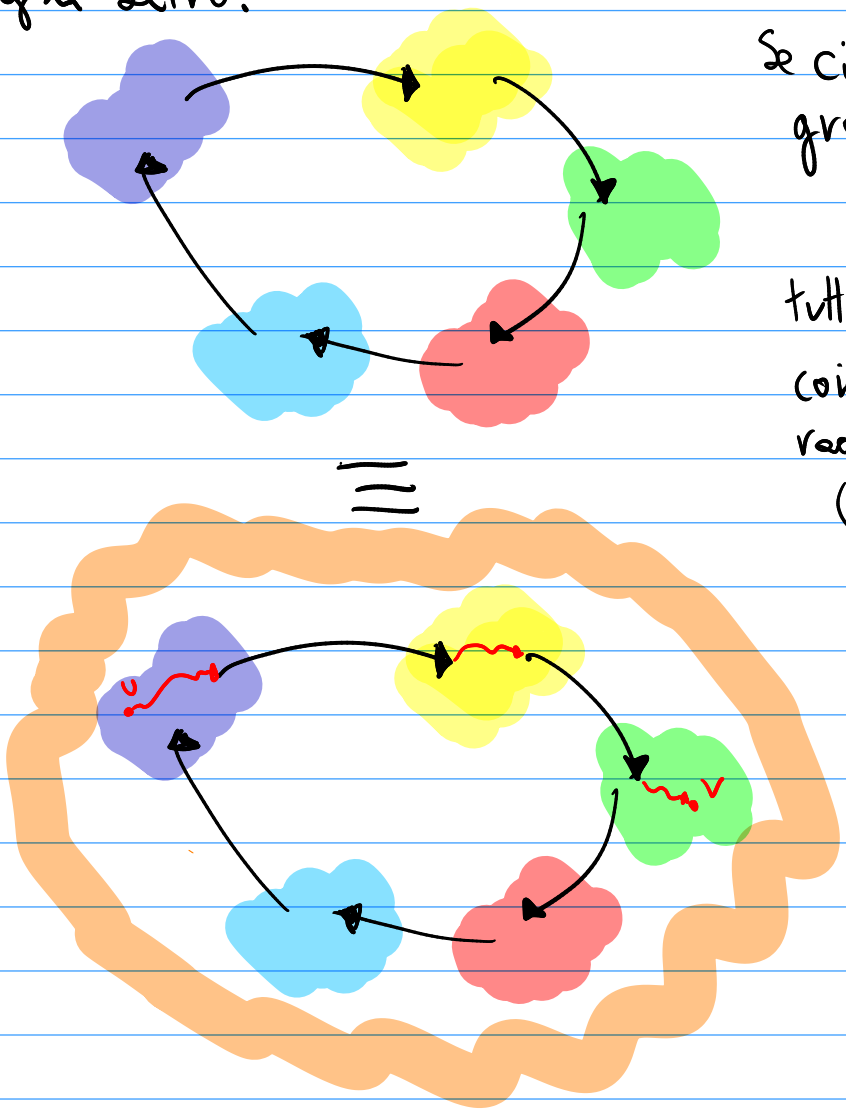
Dato G diretto, consideriamo il grafo delle sue comp. fort. conn.

- un vertice per ogni comp. connessa
- un arco dalla comp. A alla comp. B se $(u,v) \in E(G)$ con $u \in A$ e $v \in B$



Prop Il grafo delle componenti fortemente connesse è un grafo diretto aciclico (DAG)

RICORDIAMO:
dim I vertici in una componente sono raggiungibili ognuno da ogni altro.



Se ci fosse un ciclo nel grafo delle componenti

↓

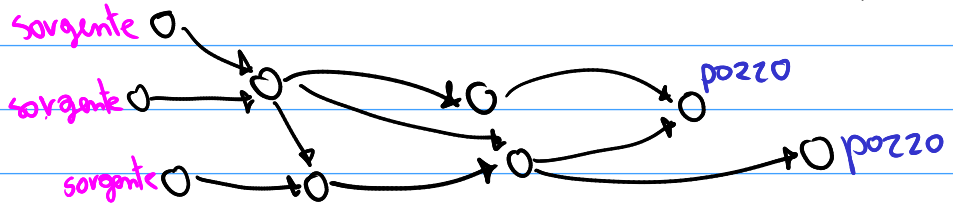
tutti i vertici delle componenti coinvolte sarebbero raggiungibili tra loro (vedi esempio U e V)

Come possiamo calcolare le componenti fortemente connesse di un grafo $G=(V,E)$?

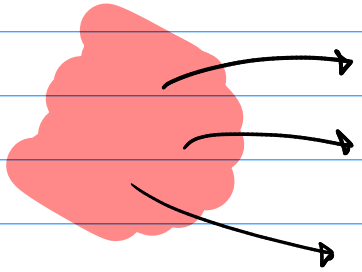
- vedremo un algoritmo basato su due chiamate a DFS, quindi di complessità $O(|V|+|E|)$

Definizioni Utili

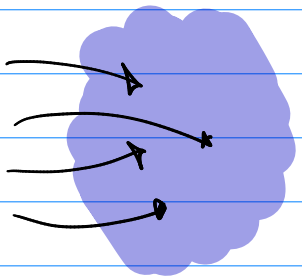
In un DAG, un vertice senza archi entranti è detto "sorgente" e uno senza archi uscenti è detto "pozzo"



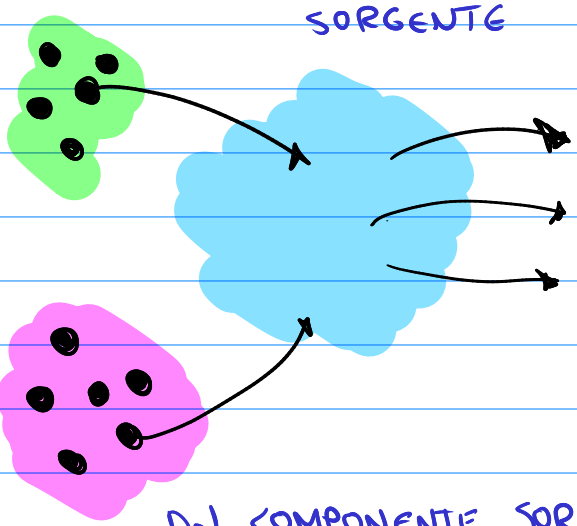
Generalizziamo la nozione alle componenti fortemente connesse

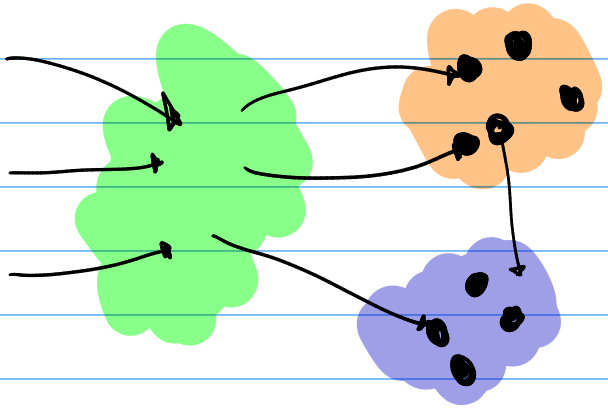


COMPONENTE SORGENTE



COMPONENTE POZZO





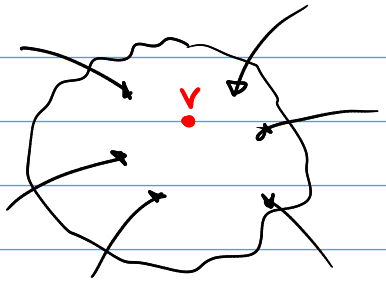
Def COMPONENTE SORGENTE AL PASSO t

"Al passo t ha solo archi entranti da componenti già visitate"

Def COMPONENTE POZZO AL PASSO t

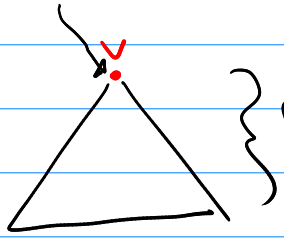
"Al passo t ha solo archi uscenti verso componenti già visitate"

- Come si comporta DFS su una componente pozzo?



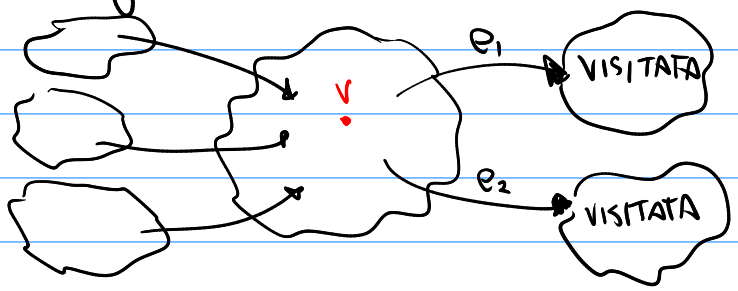
Supp. v sia il primo vertice della componente scoperto dalla visita:
- sia raggiunto da un altro vertice
- sia attraverso una chiamata DFS(G,v)

per il teorema del cammino bianco tutti i vertici della componente vengono visitati e saranno discendenti di v nell'albero DF

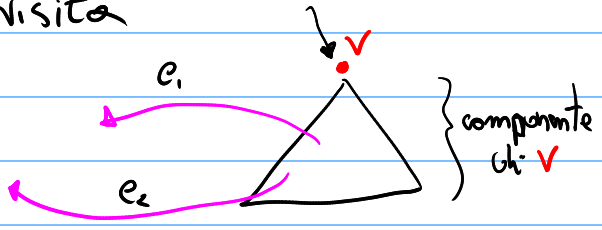


la componente fortemente connessa di v

- Analogamente...



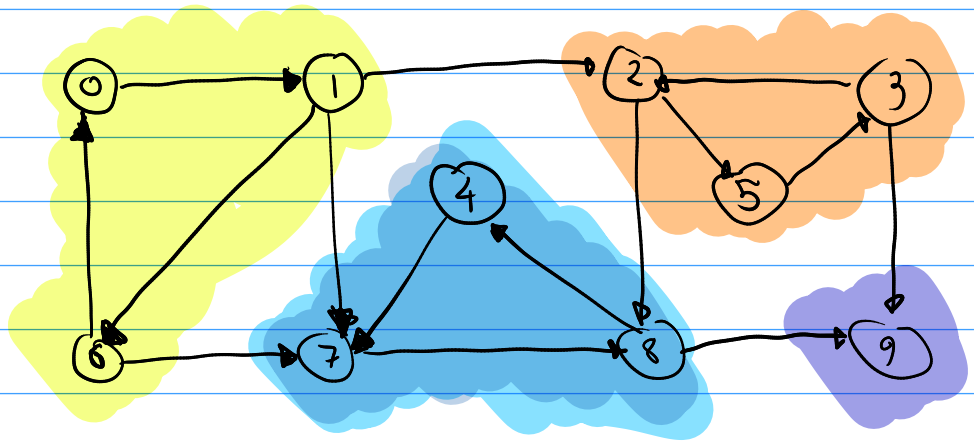
Il discorso fatto vale anche quando si visita il primo vertice di una componente che è un pozzo in quel momento della visita



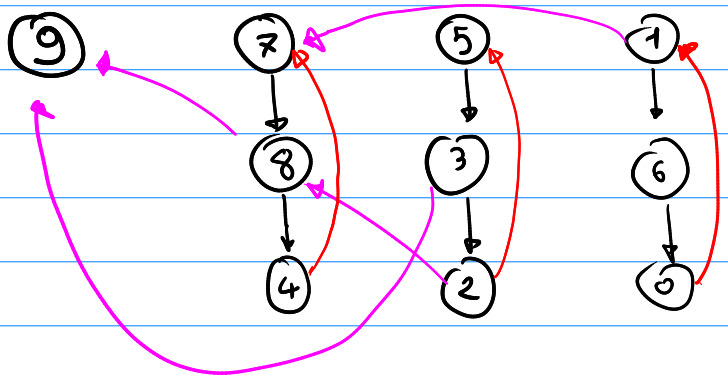
e_1, e_2 sono CROSSING EDGES

- Questo suggerirebbe una strategia possibile
  1. trovare un vertice v qualunque in una componente
     - non sia stata visitata
     - in quel momento è una componente pozzo
  2. eseguire la visita a partire da v

Ad esempio
- visito 9
- visito 7
- visito 5
- visito 1



Albero DF:



Ogni albero nella visita è una componente fortemente connessa.

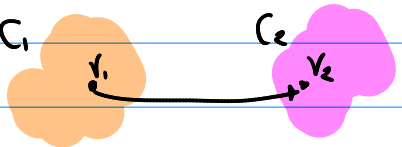
PROBLEMA: NON SAPPIAMO TROVARE EFFICIENTEMENTE UN VERTICE NELLA PROSSIMA COMPONENTE POZZO DA ANALIZZARE

tuttavia... possiamo trovare un vertice di una delle componenti SORGENTE

Def Se C è una componente fortemente connessa definiamo $f(C) = \max_{v \in C} fine[v]$

Lemma Siano C_1 e C_2 due componenti fortemente connesse di G, con un arco (v_1, v_2) e $v_1 \in C_1$, $v_2 \in C_2$

allora $f(C_1) > f(C_2)$



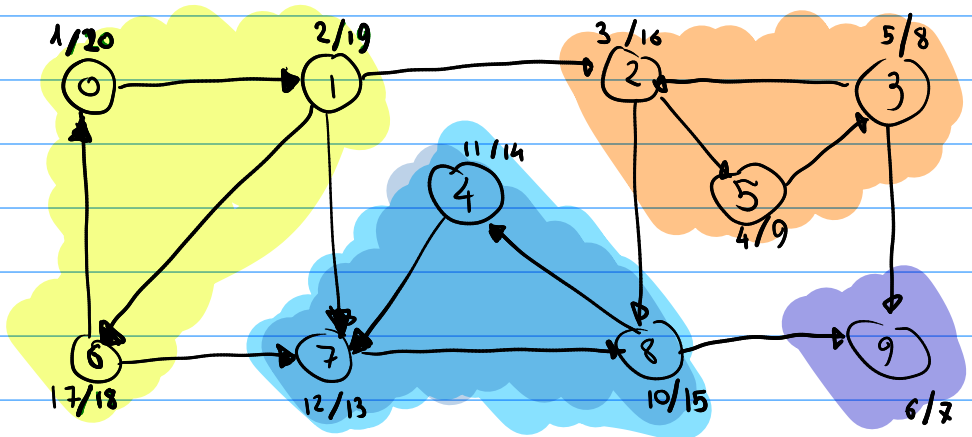
Lemma All'inizio della visita tutti i vertici in C_1 e C_2 sono bianchi

- Se il primo vertice visitato è $w_1 \in C_1$, per il teorema dei cammini bianchi tutti i vertici di C_1 e C_2 sono discendenti di w_1
  quindi $fine[w_1] > fine[v] \quad \forall v \in C_1 \cup C_2 / \{w_1\}$

- Se è invece $w_2 \in C_2$, allora $inizio[w_2] < inizio[w_1]$ e poiché w_1 non è raggiungibile da w_2, w_1 non è un discendente di w_2 nell'albero DF, mentre per il teorema del cammino bianco lo sono tutti gli elementi di C_2
  - Per il teorema delle parentesi $inizio[w_2] < fine[w_2] < inizio[w_1] < fine[w_1]$
  e $\forall v \in C_2 \quad fine[v] < fine[w_2] \rightarrow fine(w_1) > f(C_2)$

- Sappiamo, quindi, trovare un vertice in una componente sorgente

DOMANDA: Come lo troviamo?



ordine: 0, 1, 6, 2, 8, 4, 7, 5, 3, 9 (vertici ordinati per fine[v] decrescente)

DOMANDA: Non possiamo usare un metodo simile per trovare un vertice in una componente pozzo, che sarebbe quello che ci serve?

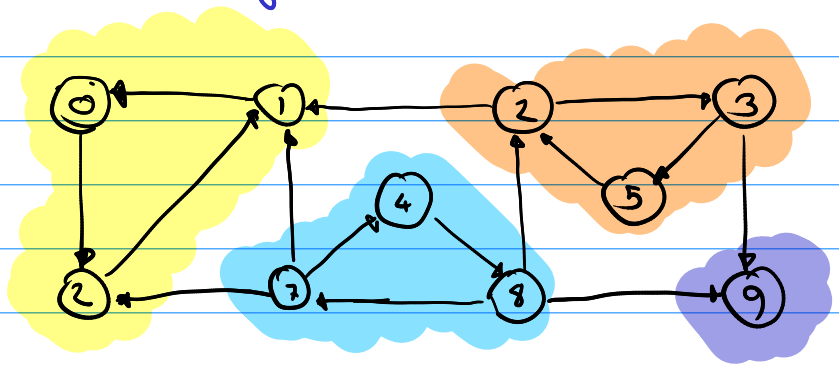
Idea Consideriamo il grafo G^T (G trasposto)

ovvero $G^T = (V, E^T)$ dove $E^T = \{(v,u), \text{ per } (u,v) \in E(G)\}$

Esercizio Dimostrare che le componenti fortemente connesse di G e G^T sono le stesse.

Osservazione le componenti pozzo di G diventano componenti sorgente di G^T e viceversa

G^T:



ordine: 0, 1, 6, 2, 8, 4, 7, 5, 3, 9

Visualizza esempio precalcolato

CONTINUA A PAG. 8

Algoritmo per il calcolo delle componenti fortemente connesse

input: $G=(V,E)$ grafo diretto output: $C_1, C_2, \dots C_k$ componenti connesse

1. Visita DFS(G) con vertici ordinati arbitrariamente
2. Sia $\langle v_1, v_2, \dots v_n \rangle$ la sequenza di vertici ordinati per $fine[v_i]$ in modo decrescente
3. Produci le liste di adiacenza di G^T
4. $color[v_i] \leftarrow$ white $\forall i$
5. for i in 1..h:
   if $color[v_i]$ = white
   $DFS(G^T, v_i)$
6. Output gli alberi della visita DF

Complessità

1. $O(|V|+|E|)$
2. si può fare durante il passo 1 senza costi aggiuntivi
3. $O(|V|+|E|)$
4. $O(|V|)$
5. è una visita DFS quindi $O(|V|+|E|)$
6. $O(|V|)$

totale $O(|V|+|E|)$

Abbozzo della dimostrazione di correttezza

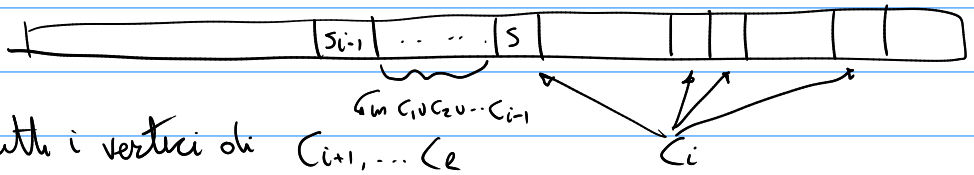
- Dopo la prima visita abbiamo $v_1, \dots, v_n$ con $fine[v_1] > fine[v_2] > \cdots fine[v_n]$
- Consideriamo le componenti connesse di G, $C_1, C_2, \dots, C_\ell$ con $f(C_1) > f(C_2) > \cdots f(C_\ell)$

- Nel ciclo 5, consideriamo i vertici $s_1, s_2, s_3, \dots$ per i quali il ciclo esegue la visita.
- Dimostriamo per induzione che al passo i la visita di s_i produce un albero DF i cui vertici (discendenti di s_i) sono tutti e soli i vertici di C_i e $fine[s_i] = f(C_i)$

Per $i=1$ $s_1 = v_1$ e $v_1 \in C_1$. Non ci sono archi uscenti da C_1 verso altre componenti connesse e quindi come discusso a pag. 4, i discendenti di s_1 sono i vertici di C_1

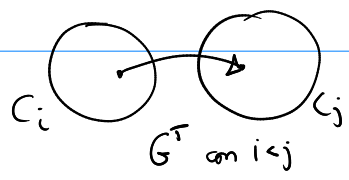
Per $i>1$ Per ipotesi induttiva $s_1, s_2, \dots s_{i-1}$ sono le radici degli alberi DF ottenuti fino al passo $i-1$, contenenti i vertici di $C_1, \dots C_{i-1}$

- Sia s = il vertice per cui $fine[s] = f(C_i)$
- Tutti i vertici di C_i sono, nell'ordine, successivi a s_{i-1} e s è il primo di essi

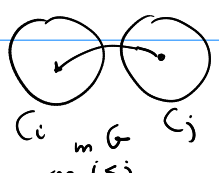


- Tutti i vertici di $C_{i+1}, \dots C_\ell$ sono successivi a s perché $f(s) > f(C_{i+1}) > f(C_{i+2}) \cdots$
- Quindi s è scelto come s_i e quindi $fine[s_i] = f(C_i)$

- La visita in G^T a partire da s_i raggiunge tutti i vertici di C_i perché sono tutti bianchi a questo punto
- La visita non tocca $C_1 \cdots C_{i-1}$ perché sono tutti neri, a questo punto
- La visita non tocca $C_{i+1}, C_{i+2}, \dots C_\ell$ perché se ci fosse un arco



allora



ma allora $f(C_j) > f(C_i)$ [vedi pag. 5] che contraddice $f(C_1) > f(C_2) > \cdots > f(C_\ell)$

Esercizio 22.5-1

- Dato un grafo non orientato $G = (V, E)$ se aggiungiamo un arco a G, di quanto può cambiare il numero di componenti connesse
- Dato un grafo ORIENTATO, di quanto può cambiare il numero di componenti FORTEMENTE CONNESSE, se aggiungiamo un arco?

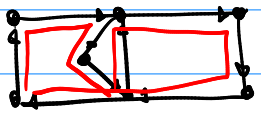
Esercizio 22.3

- Dato un grafo orientato $G = (V, E)$
- Un CIRCUITO EULERIANO, è una sequenza

$$v_1 e_1 v_2 e_2 v_3 e_3 \dots v_k e_k v_1$$

nella quale
- $e_1, e_2, \dots e_k$ sono tutti gli archi di G, senza ripetizioni
- $e_i = (v_i, v_{i+1})$

E.g.



[1] Dimostrare che $G = (V, E)$ ha un circuito euleriano se e solo se

$$\text{indeg}(v) = \text{outdeg}(v) \qquad \forall v \in V$$

[2] Mostrate un algoritmo che ne trova uno in tempo $\Theta(|E|)$