

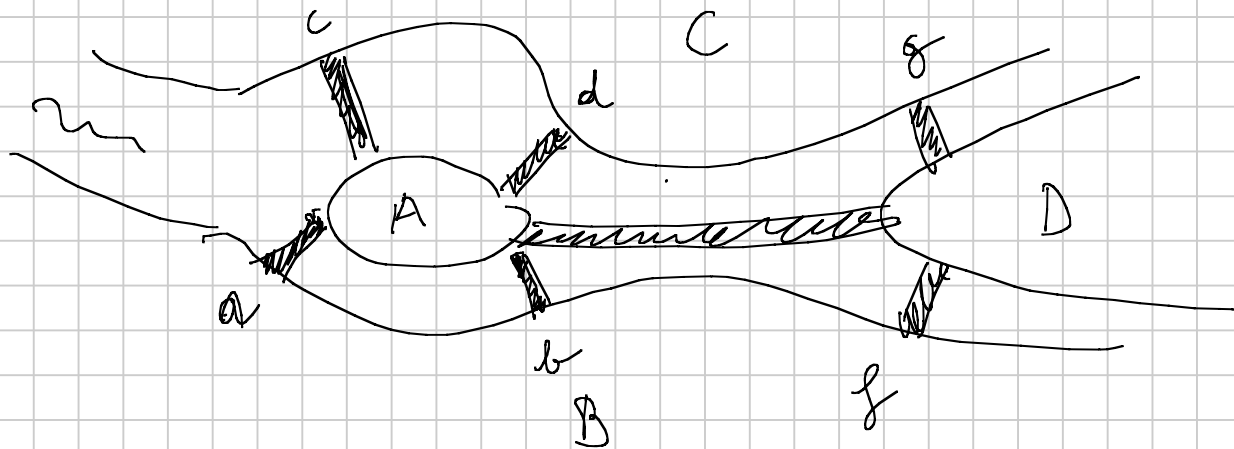
GRAFI

EULERO

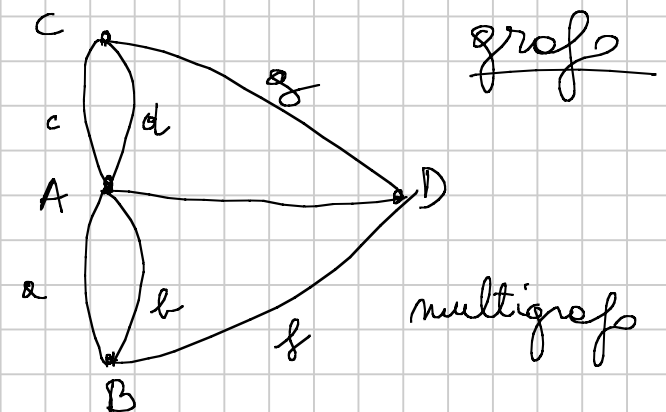
- definizioni
- memorizzazione
- Algoritmi di base

Università a Königsberg
(Kalininingrad)

Albertine

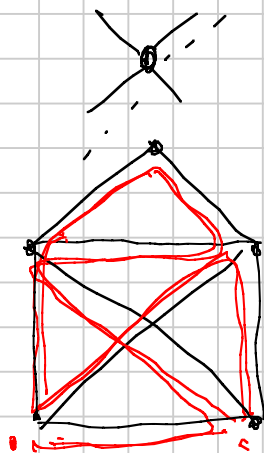


Euler Tour



Condizione necessaria e sufficiente perché esista un
Euler Tour è che i vertici abbiano grado pari

tempo polinomiale



grado (4)

Hamiltonian cycle
"difficile"

$$G = \{V, E\}$$

$$V = v_1 \dots v_n \quad |V| = n$$

$$E = \{(v_i, v_j) (v_k, v_l) \dots ()\} \quad |E| = m$$

$V =$
 $\swarrow$ vertici
 $\searrow$ modi

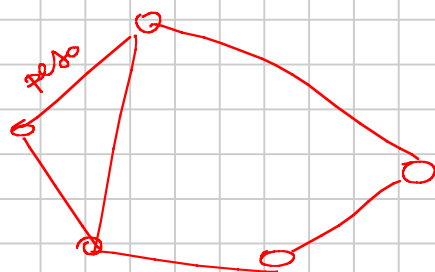
$E =$
 $\swarrow$ spigoli
 $\searrow$ archi

Edge

G è orientato se (v_i, v_j)
 (directed)

G non orientato

G è pesato



$$G = \{V, E, W\}$$

$$W : E \rightarrow \mathbb{R}$$

n, m : albero

n, m : grafo

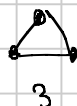
Grafo : $|V| = n$ $|E| = ?$
 connesso

numero minimo di archi ? $n-1$
 di un grafo connesso

$$n-1 \leq |E| \leq \frac{n(n-1)}{2}$$

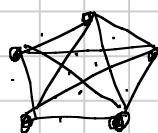
albero

$$\Theta(n)$$



graf sparso

grafo completo
 $\Theta(n^2)$



5-clique

graf denso



$$\frac{n(n-1)}{2} = \Theta(n^2) \text{ archi}$$

$\exists$ arco (u, v) in $G \Rightarrow u, v$ sono adiacenti
 $u \text{ " } (\overrightarrow{u, v}) \text{ " in } G \Rightarrow u \text{ è incidente in } v$

grado (x) = n° di archi adiacenti
 n° di archi incidenti se G è orientato

$$\sum (\text{gradi di tutti i nodi}) = 2m$$

cammino

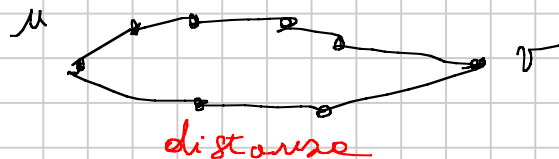
$$x_0, x_1, \dots, x_k : \exists (x_i, x_{i+1}) \in E$$



lunghezza del cammino = numero di archi (se G non è pesato)

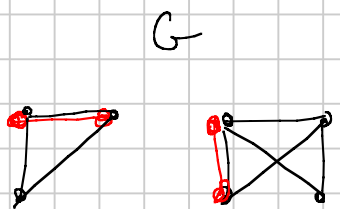
oppure = somma dei pesi degli archi (se G è pesato)

distance $(u, v) =$ lunghezza del cammino minimo



G è connesso altrimenti si dice formato di componenti
connesse

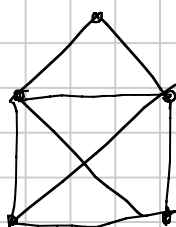
G'



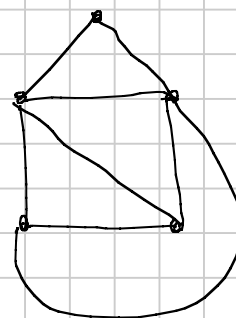
2 componenti
connesse

sottografo di $G = (V, E)$
è un grafo $G' = (V', E')$:
 $V' \subseteq V$ e $E' \subseteq E$

grafo planare se può essere disegnato senza incroci

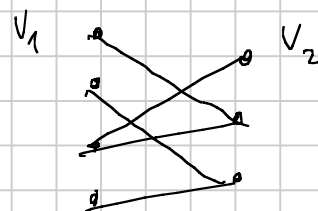


$\Rightarrow$



SI

grafo bipartito se i suoi vertici V possono essere suddivisi in V_1 e V_2 in modo tale che tutti gli archi di G vanno da V_1 a V_2

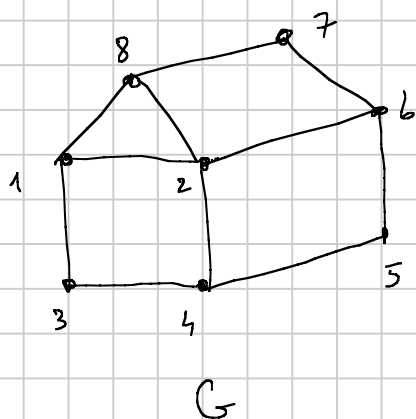


Memorizzazione

Titolo nota

17/03/2016

Matrice di adiacenze



	1	2	3	4	5	6	7	8
1		1	1					1
2	1			1		1		1
3	1			1				
4		1	1		1			
5				1		1		
6		1			1		1	
7						1		1
8	1	1					1	

$n \times n$

$$(i, i)$$

$$(i, j) = (j, i)$$

se G è orientato

$$(i, j) \neq (j, i)$$

(u, v) sono adiacenti?
grado(u)?

$$\Theta(1)$$

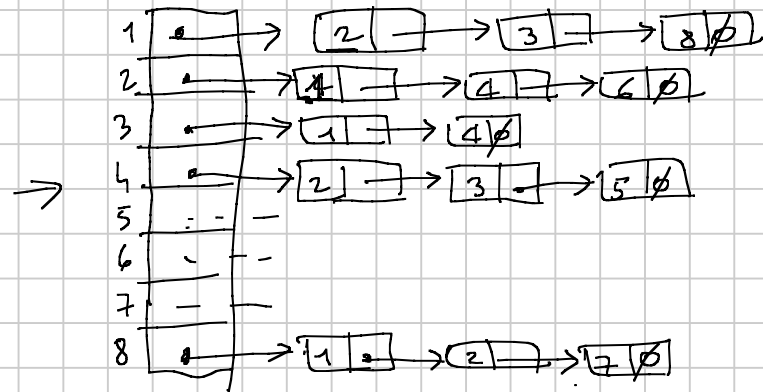
$$\Theta(n)$$

spazio $\Theta(n^2)$ bit
se G non è pesato

• LISTE D'ADIACENZA

Titolo nota

17/03/2016



$$n + l_1 + l_2 + \dots + l_n = O(n^2)$$

$\downarrow$
 $O(n)$

$\downarrow$
 $O(n)$

n° di elementi di liste complessive = $(2m)$

spazio = $\Theta(n + m)$

lineare con n, m

grafo statico e liste di adiacenze

VISITE

BFS

DFS

BFS

Breadth First Search

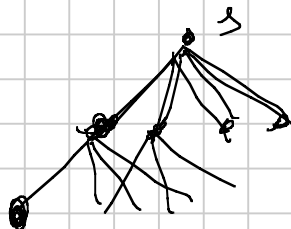
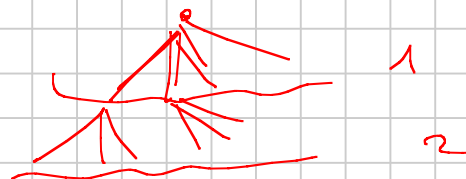
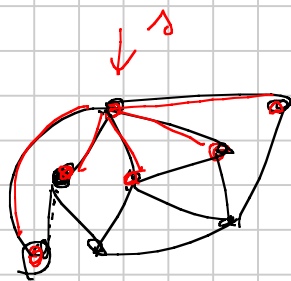
DFS

Depth First Search

Visita in ampiezza
Visita in profondità

BFS

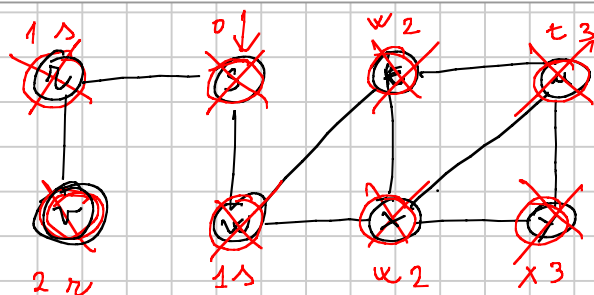
equivale a
una visita per
livelli in un
albero



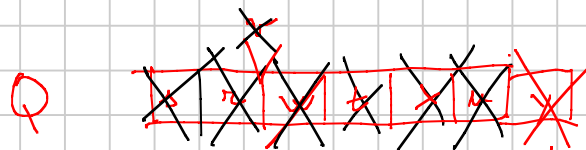
CODA

$BFS(G, s)$
 for ogni vertice $u \in V - \{s\}$ {
 $u.colore = bianco$;
 $u.\pi = NIL$;
 $u.d = \infty$;
 }
 $s.colore = grigio$; $s.\pi = NIL$; $d = 0$;
 $Q = \emptyset$;
 $ENQUEUE(Q, s)$;
 while ($Q \neq \emptyset$) { $u = DEQUEUE(Q)$;
 for ogni vertice $v \in Adj[u]$ {
 if ($v.colore == bianco$) { $v.colore = grigio$; $v.d = u.d + 1$; $v.\pi = u$;
 $ENQUEUE(Q, v)$;
 }
 }
 $u.colore = nero$; ?

$x.colore \leftarrow \begin{matrix} bianco & = & mai\ considerato \\ grigio & = & iniziata\ la\ visita \\ nero & = & terminata\ la\ visita \end{matrix}$
 $x.\pi = \text{predecessore nella visita}$
 $x.d = \text{calcolo la distanza a } s$
 $Q = \text{code}$



$s \rightarrow \underline{s}, \underline{r}$
 $r \rightarrow \underline{r}, \underline{w}$
 $t \rightarrow \underline{u}, \underline{w}, \underline{x}$
 $u \rightarrow \underline{t}, \underline{x}, \underline{y}$
 $v \rightarrow \underline{r}$
 $w \rightarrow \underline{s}, \underline{t}, \underline{x}$
 $x \rightarrow \underline{t}, \underline{u}, \underline{w}$
 $y \rightarrow \underline{u}, \underline{x}$



Sequenze di visita:

s, r, w, v, t, x, u, y

$$\Theta(n + m)$$