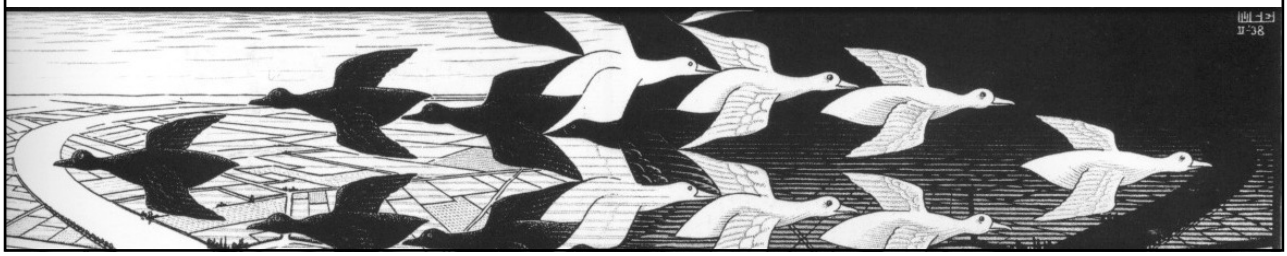
Università degli Studi Roma Tre – Dipartimento di Matematica e Fisica – Corso di Laurea in Matematica

Appunti del corso IN110 Algoritmi e Strutture Dati

7 – Grafi e alberi: introduzione

Prof. Marco Liverani

(liverani@mat.uniroma3.it – http://www.mat.uniroma3.it/users/liverani/IN110)



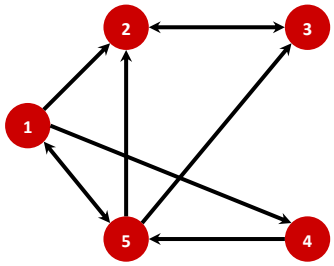
Sommario

- Grafi: definizioni principali
- Cammini, cicli
- Grafi connessi, componenti connesse di un grafo, componenti fortemente connesse
- Alberi liberi, alberi radicati, alberi ricoprenti: definizioni e proprietà
- Rappresentazione di grafi ed alberi
- Problemi ed algoritmi sui grafi
- Problemi di ottimizzazione sui grafi

Grafi: definizioni principali 1

- Un grafo G è una coppia ordinata di insiemi $G = (V, E)$, dove $V = \{v_1, \ldots, v_n\}$ è l'insieme dei vertici (o nodi, o punti) ed E è l'insieme degli spigoli (o *archi*, o *lati*)
- E è una relazione binaria su V, ovvero un sottoinsieme del prodotto cartesiano $V \times V$: $e = (v_i, v_j) \in E \subseteq V \times V$
- Se il grafo G è non orientato la relazione E è simmetrica: $(v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E$
- Se il grafo G è orientato $(v_i, v_j) \in E$ non implica necessariamente che $(v_j, v_i) \in E$
- Sia $G = (V,E)$; notazione:
  - Numero di vertici: $|V| = \boldsymbol{n}$
  - Numero di spigoli: $|E| = \boldsymbol{m}$
- Sia $G = (V, E)$ un grafo orientato; se $(u, v) \in E$ allora si dice che lo spigolo è incidente i vertici u e v; lo spigolo è uscente dal vertice u ed è entrante nel vertice v
- Il grado di un vertice è dato dal numero di spigoli incidenti su di esso
- Se $(u, v) \in E$ allora v è adiacente ad u (se il grafo è orientato allora non è detto che u sia adiacente a v)

Grafi: definizioni principali 2



$G = (V,E)$ è un grafo orientato
$V = \{1, 2, 3, 4, 5\} \Rightarrow |V| = n = 5$
$E = \{(1,2), (1,4), (1,5), (2,3), (3,2), (4,5), (5,1), (5,2), (5,3)\} \Rightarrow |E| = m = 9$

Cammini, cicli, connessione

- Un **cammino** di lunghezza k da un vertice u ad un vertice u' in G è una sequenza $(v_0, v_1, \ldots, v_k)$ di vertici tali che $u=v_0$, $u'=v_k$ e $(v_{i-1}, v_i) \in E$, per $i=1, \ldots, k$
- Se esiste un cammino p da u a u', allora si dice che u' è **raggiungibile** da u
- Un cammino è **semplice** se tutti i suoi vertici sono distinti. Un cammino da u a u' è un **ciclo** se $u=u'$; se il ciclo è di lunghezza 1 allora è un **cappio**
- Un grafo G privo di cicli è **aciclico**

- Un grafo **non** orientato è **connesso** se ogni coppia di vertici è collegata da un cammino in G
- Le **componenti connesse** di un grafo sono le classi di equivalenza dei vertici sotto la relazione «raggiungibile da»
- Un grafo orientato è **fortemente connesso** se ogni coppia di vertici è collegata da un cammino; le **componenti fortemente connesse** di un grafo sono le classi di equivalenza dei vertici sotto la relazione «mutuamente raggiungibile»

Sottografi

- Un grafo $G' = (V', E')$ è un **sottografo** di $G = (V, E)$ se $V' \subseteq V$ e $E' \subseteq E$
- Dato un insieme $V' \subseteq V$, il sottografo di $G = (V, E)$ **indotto** da V' è il grafo $G' = (V', E')$ dove $E' = \{(u,v) \in E: u, v \in V'\}$
- Un grafo **completo** è un grafo non orientato in cui ogni coppia di vertici è adiacente
- Un grafo **bipartito** è un grafo non orientato $G = (V, E)$ dove V può essere partizionato in due insiemi V_1 e V_2 tali che $(u,v) \in E \Rightarrow u \in V_1$ e $v \in V_2$, oppure $u \in V_2$ e $v \in V_1$

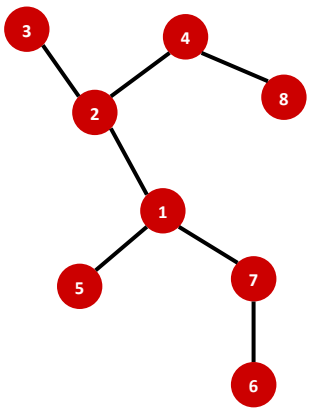
Alberi liberi

- Un **albero** è un grafo *non orientato*, *connesso* e *aciclico* (privo di cicli)
- Un grafo non orientato e aciclico, ma non connesso, è una **foresta**
- Sia $G = (V, E)$ un grafo non orientato. Affermazioni equivalenti (*Teorema di caratterizzazione degli alberi*):
  - G è un albero
  - due vertici qualsiasi di G sono connessi da un unico cammino semplice
  - G è connesso, ma se qualunque spigolo di G venisse rimosso, allora G diventerebbe non connesso
  - G è connesso e $|E| = |V| - 1$
  - G è aciclico e $|E| = |V| - 1$
  - G è aciclico, ma se venisse aggiunto uno spigolo qualsiasi il grafo risultante non sarebbe più aciclico

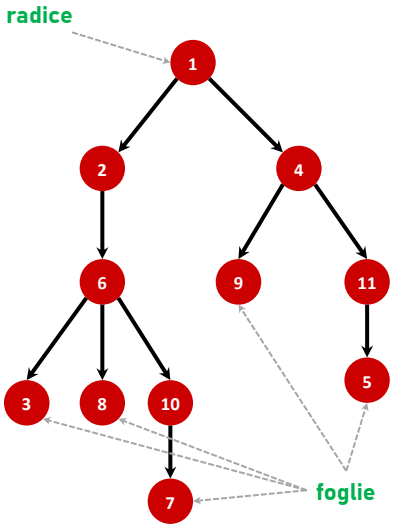
Alberi radicati

- Un **albero radicato** è un albero in cui uno dei vertici (detto **radice**) si distingue dagli altri
- Se u è un vertice dell'albero radicato A ed r è la sua radice, allora ogni vertice v sull'unico cammino da r fino a u è un **antenato** di u e u è un **discendente** di v
- Se (u,v) è uno spigolo di un albero radicato, allora si dice che u è il **padre** di v e che v è un **figlio** di u; se anche (u,w) è uno spigolo dell'albero, allora v e w sono **cugini**
- Un vertice senza figli è una **foglia** dell'albero
- La lunghezza del cammino più lungo dalla radice ad una foglia è l'**altezza** dell'albero

Alberi: esempi



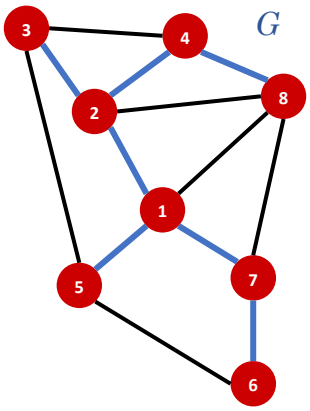
Albero (libero)

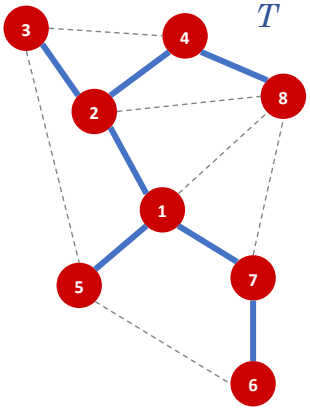


Albero radicato orientato

Alberi ricoprenti (spanning tree)

- Sia $G = (V,E)$ un grafo connesso, sia $T = (V,E')$ un sottografo di G connesso
- Se T è un albero allora si dice che T è un **albero ricoprente** (spanning tree) di G
- Un grafo G può avere anche più di un albero ricoprente
- Se G non è connesso allora l'insieme degli alberi che ricoprono le sue componenti connesse è una **foresta di alberi ricoprenti** (spanning forest)



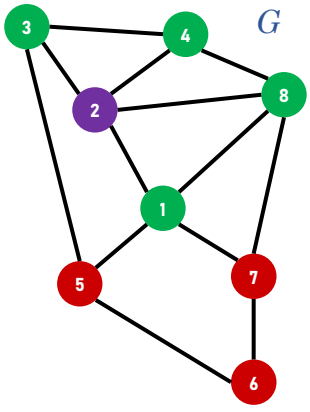


Grafo $G = (V,E)$ e un albero ricoprente $T = (V,E')$ con $E' \subset E$

Rappresentazione di grafi e alberi 1

- Nella memoria della macchina un grafo può essere rappresentato mediante:
  - Liste di adiacenza: per ogni vertice $u \in V$ viene costruita una lista dei vertici $v_1, \ldots, v_k$ adiacenti a u, ossia tali che $(u,v_i) \in E$ per ogni $i = 1, \ldots, k$
  - Matrice di adiacenza: il grafo viene rappresentato con una matrice quadrata di ordine $n = |V|$, in cui $M_{i,j} = 1$ se $(v_i,v_j) \in E$, e $M_{i,j} = 0$ altrimenti
- Per grafi *sparsi* (con pochi spigoli rispetto al numero dei vertici) è conveniente la rappresentazione mediante liste di adiacenza

Rappresentazione di grafi e alberi 2



vertici adiacenti a $v = 2$

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

liste di adiacenza di G

	1	2	3	4	5	6	7	8
1	0	1	0	0	1	0	1	1
2	1	0	1	1	0	0	0	1
3	0	1	0	1	1	0	0	0
4	0	1	1	0	0	0	0	1
5	1	0	1	0	0	1	0	0
6	0	0	0	0	1	0	1	0
7	1	0	0	0	0	1	0	1
8	1	1	0	1	0	0	1	0

matrice di adiacenza di G

Problemi e algoritmi su grafi

- Esiste un vasto insieme di problemi interessanti sui grafi, spesso dotati di algoritmi risolutori efficienti ed eleganti
- Alcuni esempi:
  - Visita in ampiezza ed in profondità di un grafo
  - Ricerca del minimo albero ricoprente (minimum spanning tree – algoritmo di Kruskal)
  - Ricerca del cammino minimo per raggiungere ogni altro vertice a partire da una sorgente (algoritmo di Dijkstra)
  - Calcolo del massimo flusso su una rete (algoritmo di Ford e Fulkerson, algoritmo di Belman e Ford, ecc.)

Problemi di ottimizzazione combinatoria

- Alcuni problemi consistono nella ricerca di una configurazione del grafo tale da massimizzare o minimizzare una determinata proprietà
- Questi problemi vanno sotto il nome di problemi di ottimizzazione combinatoria
- Formalmente viene definita una funzione obiettivo f i cui valori sono calcolati sulla base di particolari configurazioni del grafo (es.: un peso associato ai vertici o agli spigoli del grafo, il numero di vertici in ogni componente, ecc.); si deve individuare una configurazione del grafo tale da rendere massimo o minimo il valore di f