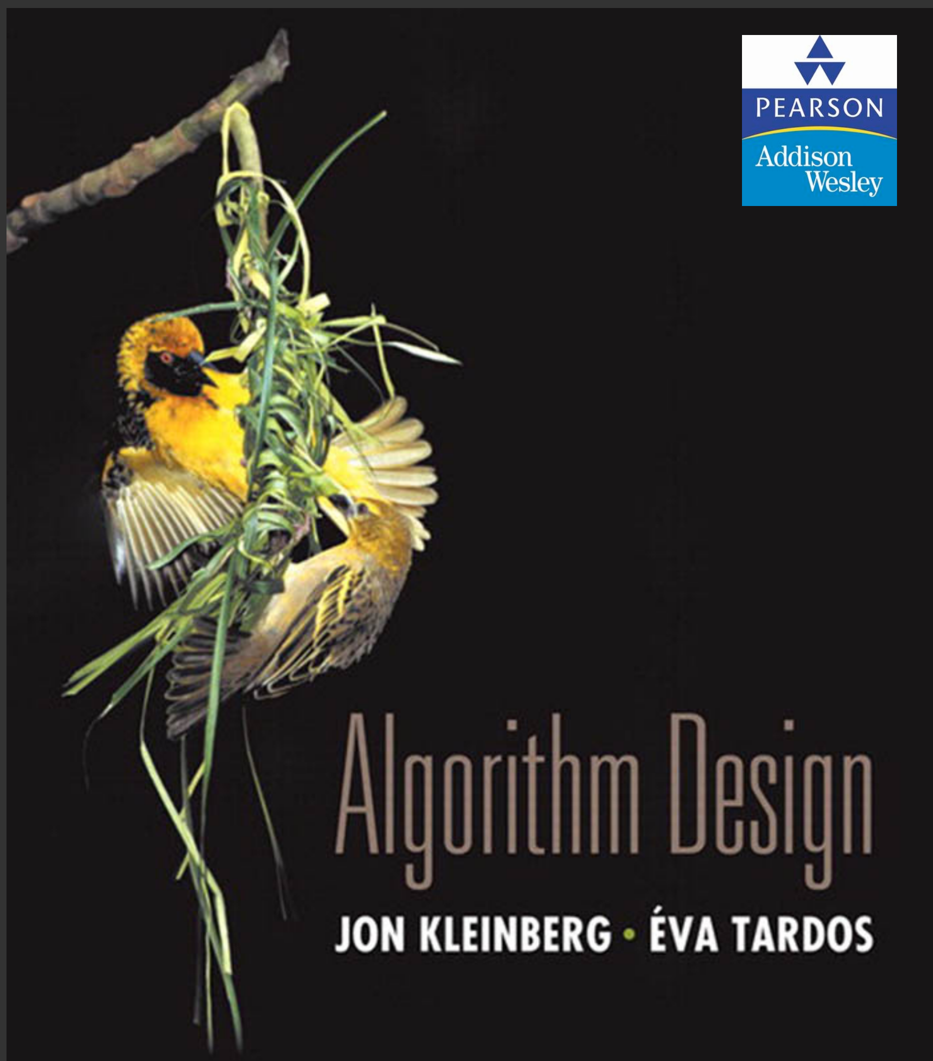


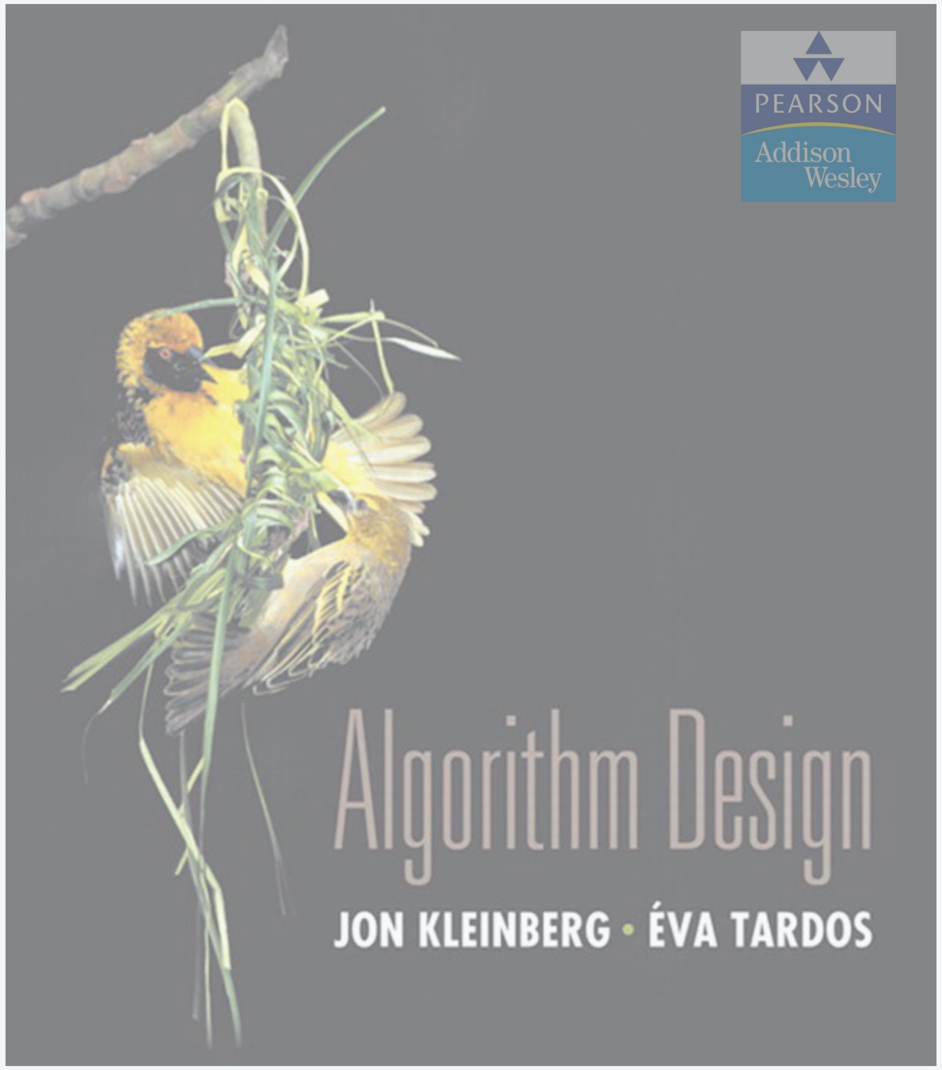
Traduzione e adattamento di Vincenzo Bonifaci
Original lecture slides by Kevin Wayne

http://www.cs.princeton.edu/~wayne/kleinberg-tardos

3. Grafi

- *definizione di base e applicazioni*
- *connettività e visite di grafi*
- *grafi bipartiti*
- *connettività nei digrafi*
- *DAG e ordinamento topologico*

Last updated on 04/03/2024 18:05



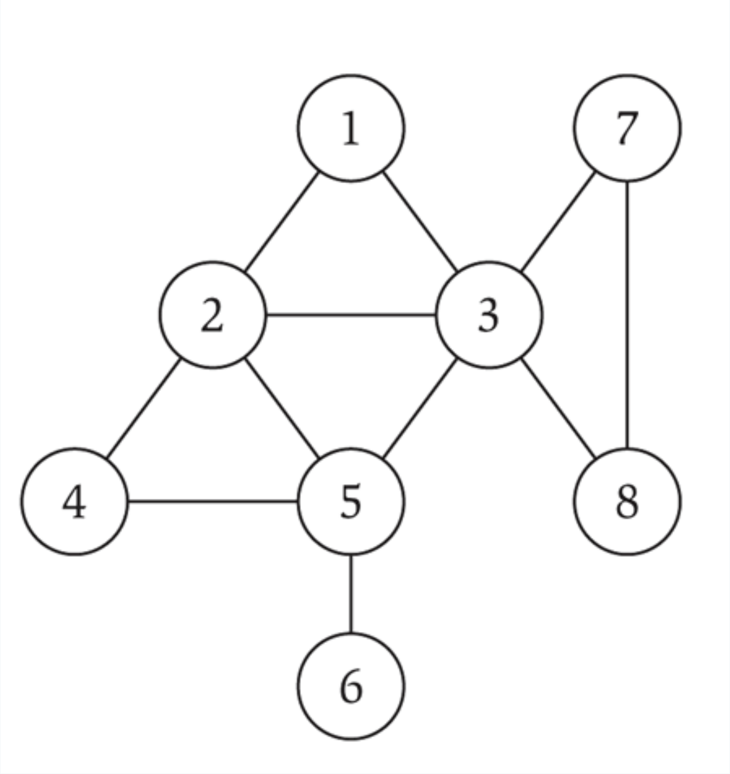
3. Grafi

- *definizioni di base e applicazioni*
- *graph connectivity and graph traversal*
- *testing bipartiteness*
- *connectivity in directed graphs*
- *DAGs and topological ordering*

Grafi (non orientati) [undirected graphs]

Notazione. $G = (V, E)$

- V = nodi (o vertici).
- E = archi (o spigoli) tra coppie di nodi.
- Cattura una relazione tra coppie di oggetti.
- Parametri di taglia del grafo: $n = |V|, m = |E|$.

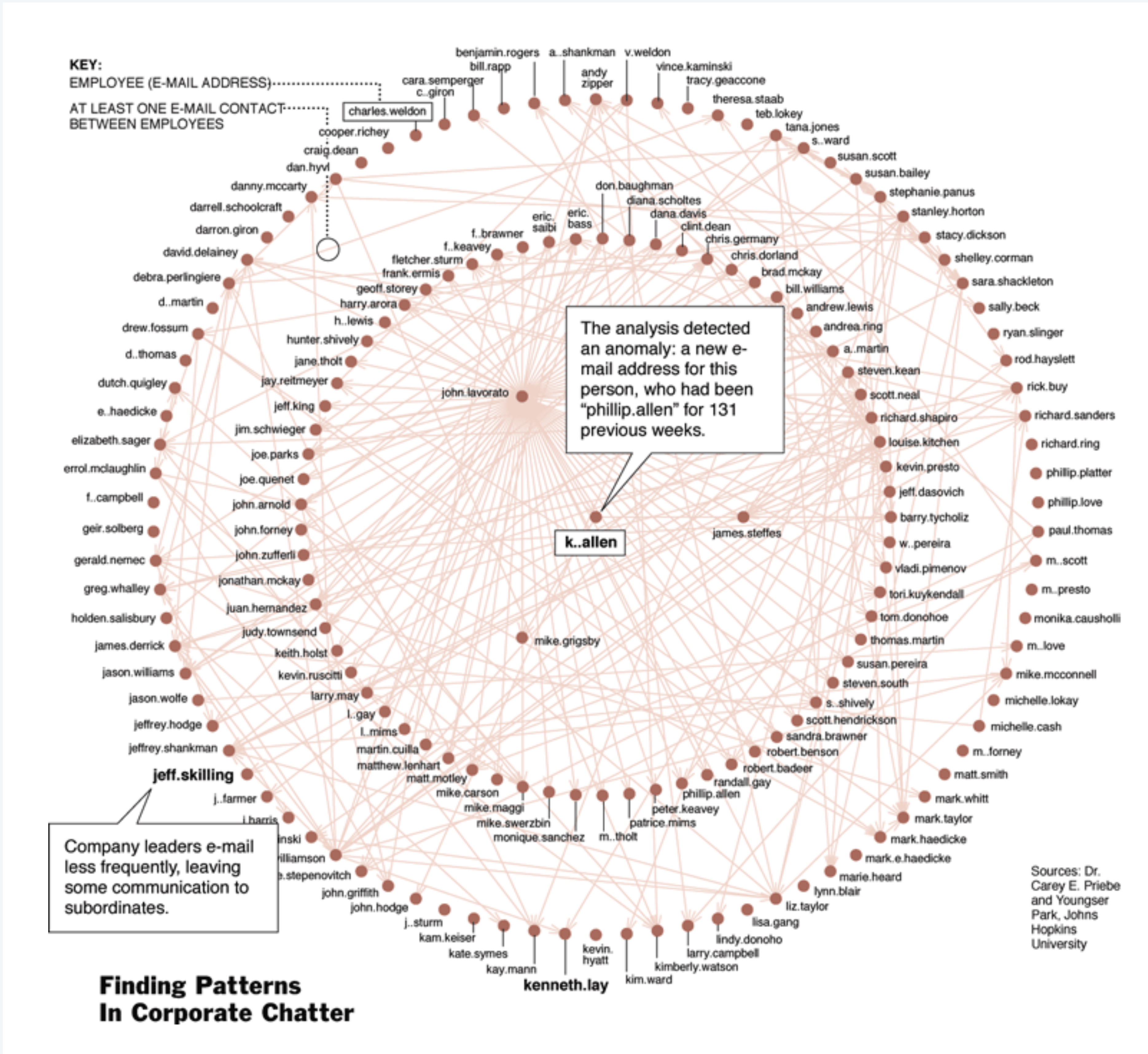


$V = \{ 1, 2, 3, 4, 5, 6, 7, 8 \}$

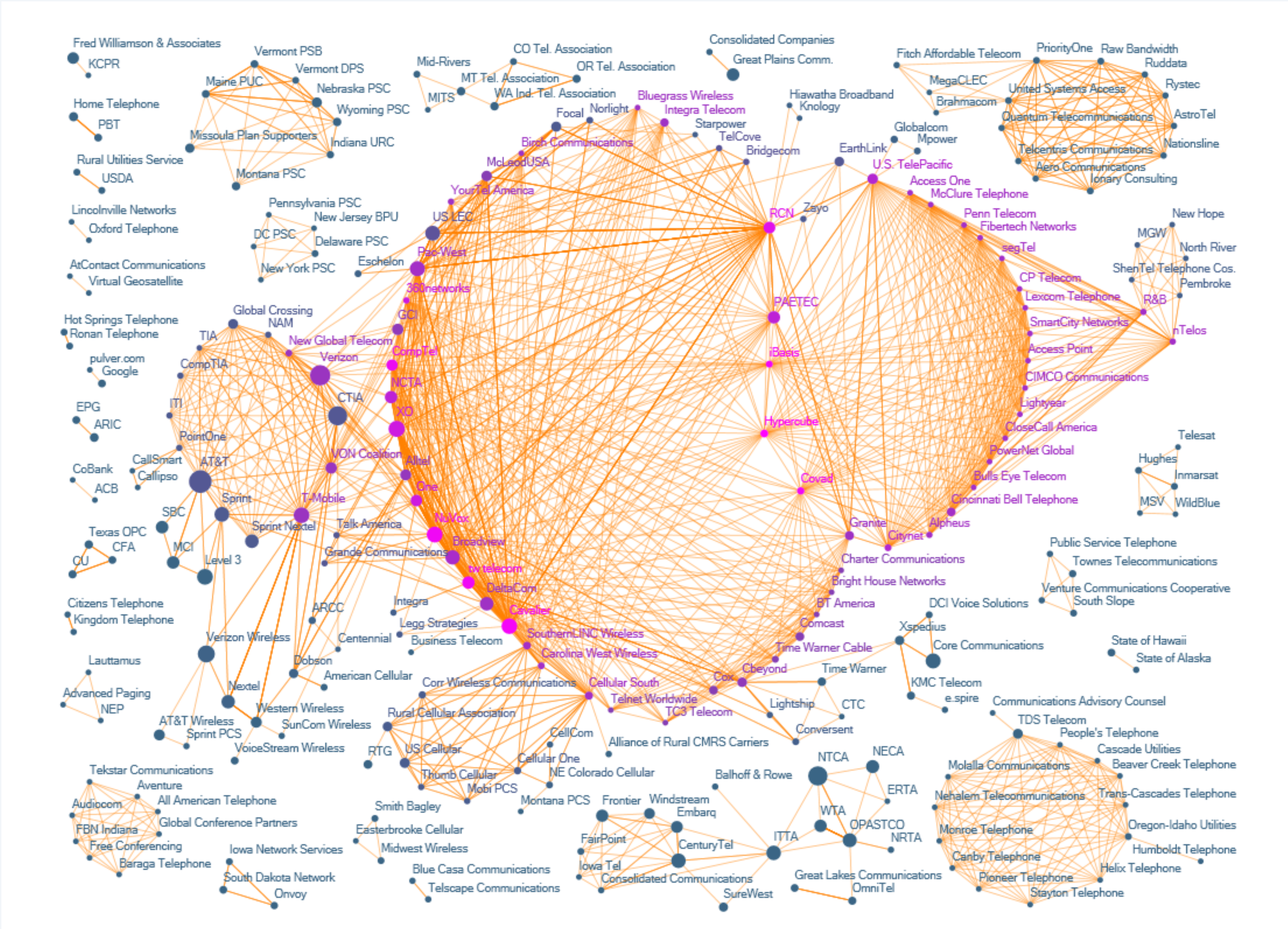
$E = \{ 1–2, 1–3, 2–3, 2–4, 2–5, 3–5, 3–7, 3–8, 4–5, 5–6, 7–8 \}$

$m = 11, n = 8$

Una settimana di email Enron



Evoluzione delle coalizioni lobbistiche della Commissione FCC



"The Evolution of FCC Lobbying Coalitions" by Pierre de Vries in JoSS Visualization Symposium 2010

Framingham heart study

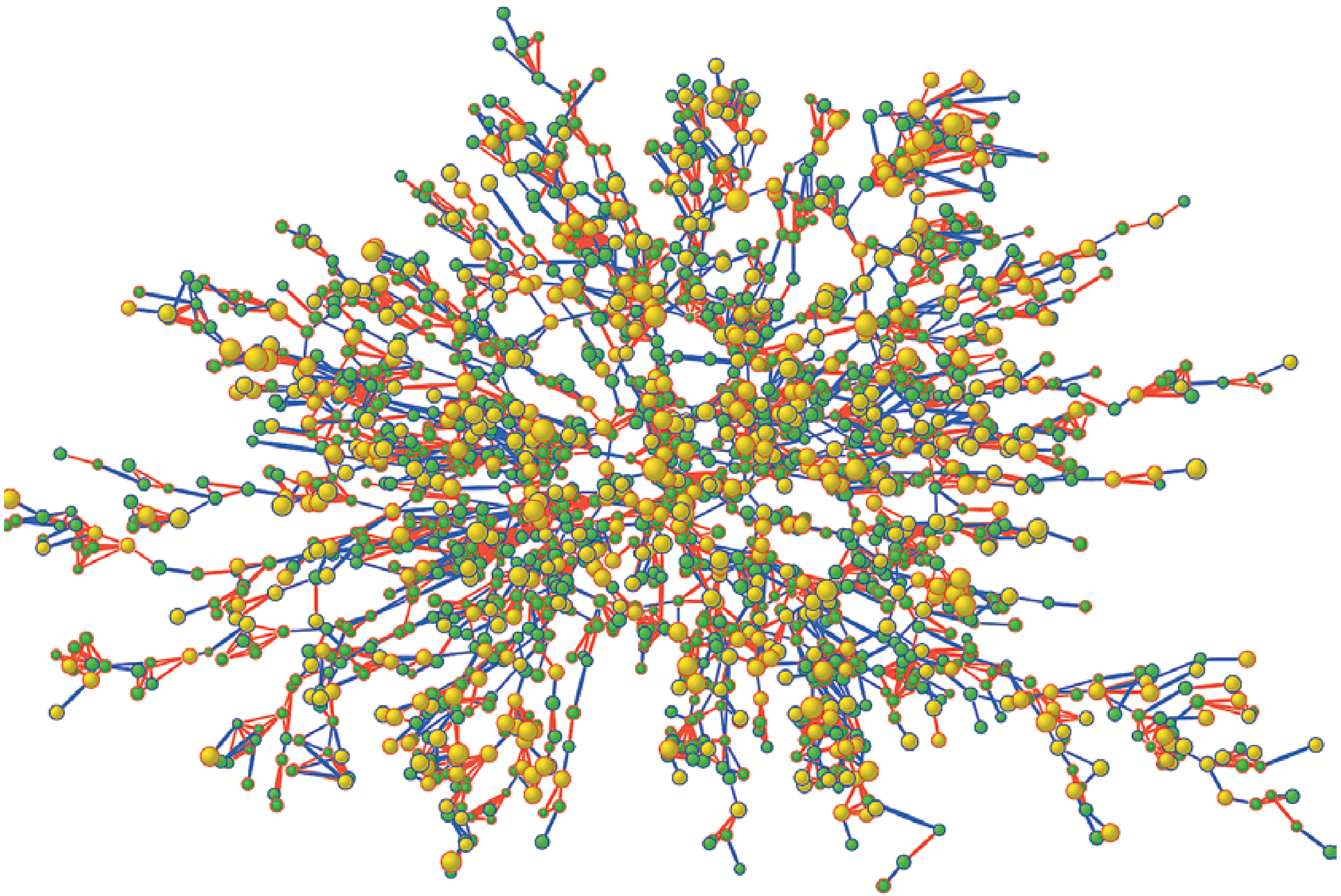


Figure 1. Largest Connected Subcomponent of the Social Network in the Framingham Heart Study in the Year 2000.
Each circle (node) represents one person in the data set. There are 2200 persons in this subcomponent of the social network. Circles with red borders denote women, and circles with blue borders denote men. The size of each circle is proportional to the person's body-mass index. The interior color of the circles indicates the person's obesity status: yellow denotes an obese person (body-mass index, ≥30) and green denotes a nonobese person. The colors of the ties between the nodes indicate the relationship between them: purple denotes a friendship or marital tie and orange denotes a familial tie.

"The Spread of Obesity in a Large Social Network over 32 Years" by Christakis and Fowler in New England Journal of Medicine, 2007

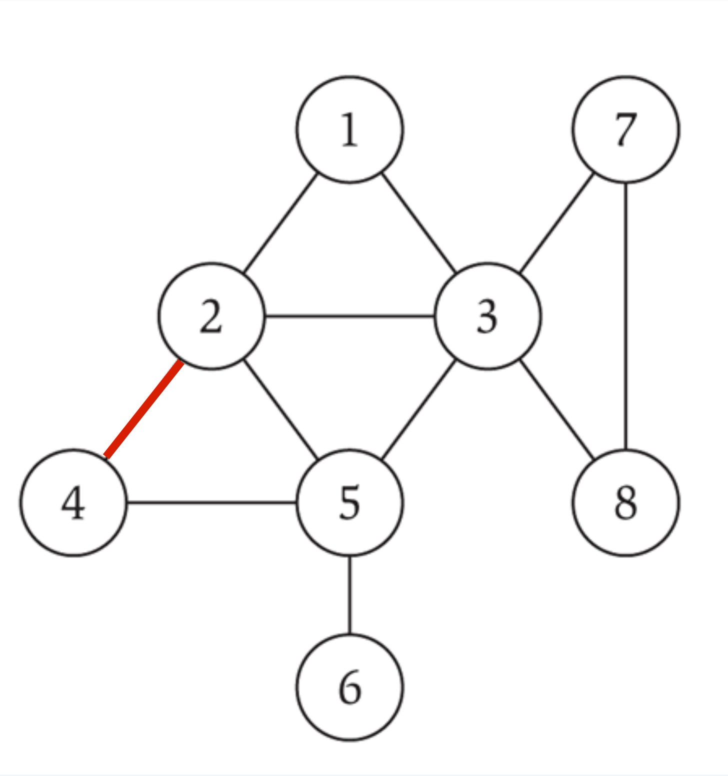
Applicazioni dei grafi

grafo	nodo	arco
di comunicazione	telefono, computer	cavo in fibra ottica
circuitale	porta, registro, processore	collegamento elettrico
meccanico	giunto	asta, trave, molla
finanziario	titolo azionario, moneta	transazioni
di trasporto	intersezione di strade, aeroporto	autostrada, tratta aerea
di internet	rete di classe C	connessione
di un gioco	situazione del tabellone di gioco	mossa lecita
relazione sociale	persona, attore	amicizia, copresenza nel cast
rete neurale	neurone	sinapsi
rete di proteine	proteina	interazione proteina-proteina
molecolare	atomo	legame

Rappresentazione di grafi: matrice di adiacenza

Matrice di adiacenza. Matrice $n \times n$ con $A_{uv} = 1$ se (u, v) è un arco.

- Due rappresentazioni di ogni arco.
- Spazio proporzionale a n^2.
- Verificare se (u, v) è un arco richiede tempo $\Theta(1)$.
- Identificare tutti gli archi richiede tempo $\Theta(n^2)$.



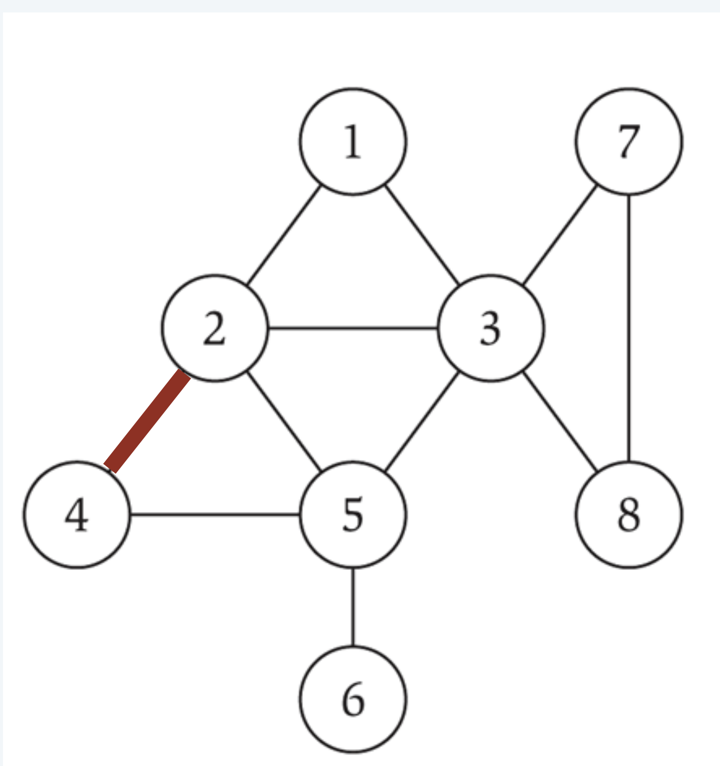
	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

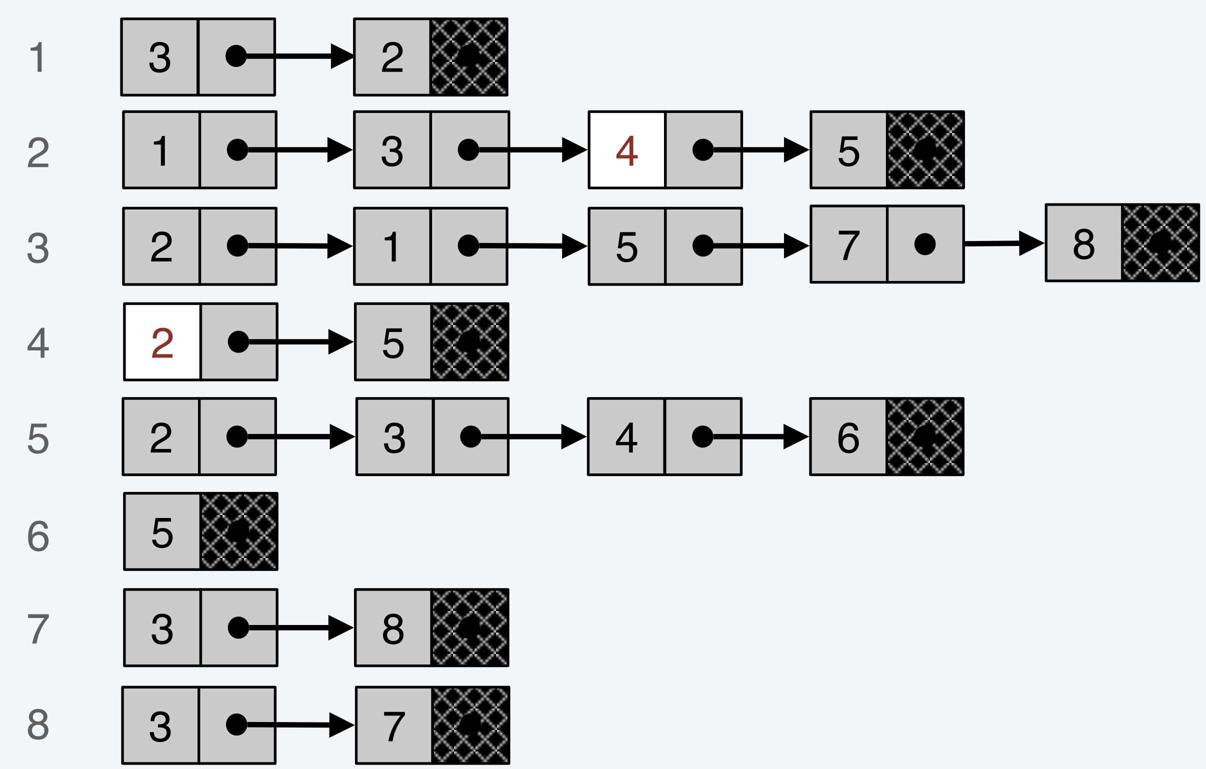
Rappresentazione di grafi: liste di adiacenza

Liste di adiacenza. Un array, indicizzato da nodi, di liste.

- Due rappresentazioni di ogni arco.
- Spazio $\Theta(m + n)$.
- Verificare se (u, v) è un arco richiede tempo $O(grado(u))$.
- Identificare tutti gli archi richiede tempo $\Theta(m + n)$.

grado = numero di vicini di u



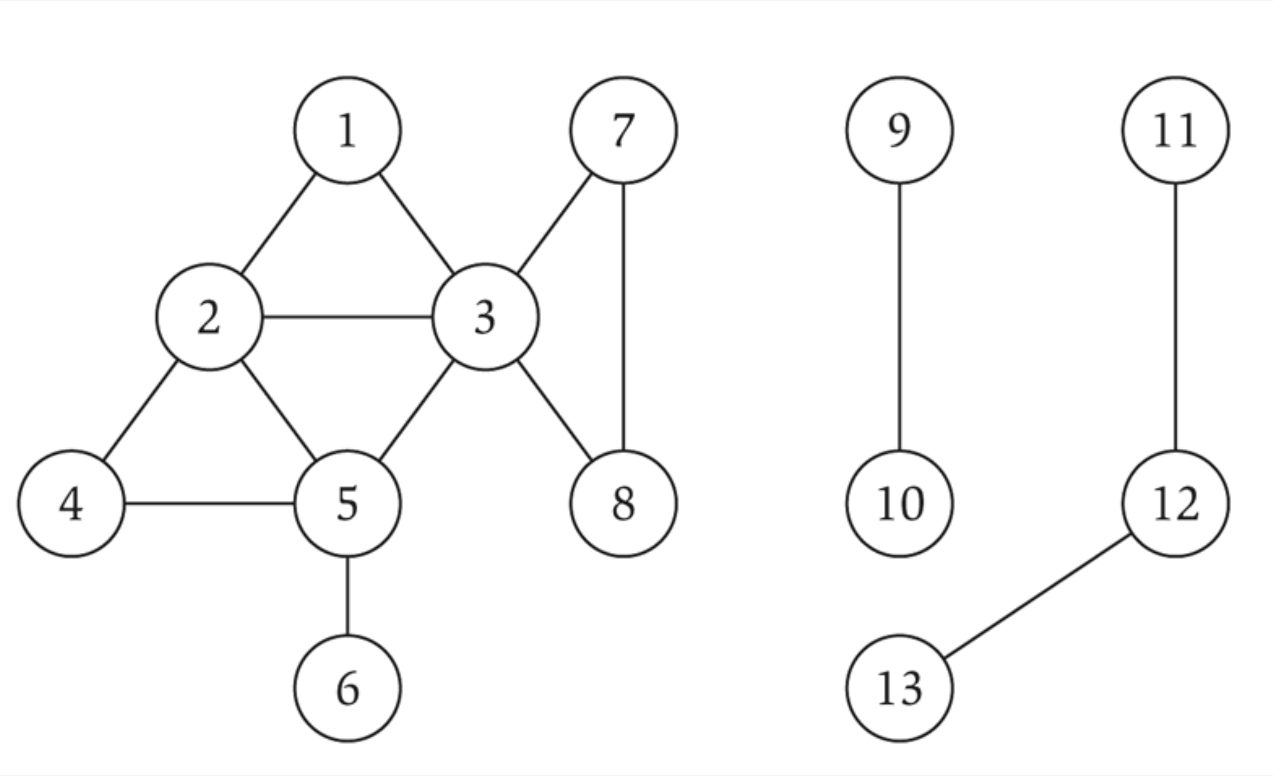


Cammini e connettività

Def. Un cammino in un grafo non orientato $G = (V, E)$ è una sequenza di nodi v_1, v_2, …, v_k con la proprietà che ciascuna coppia consecutiva v_{i-1}, v_i è collegata da un arco di E.

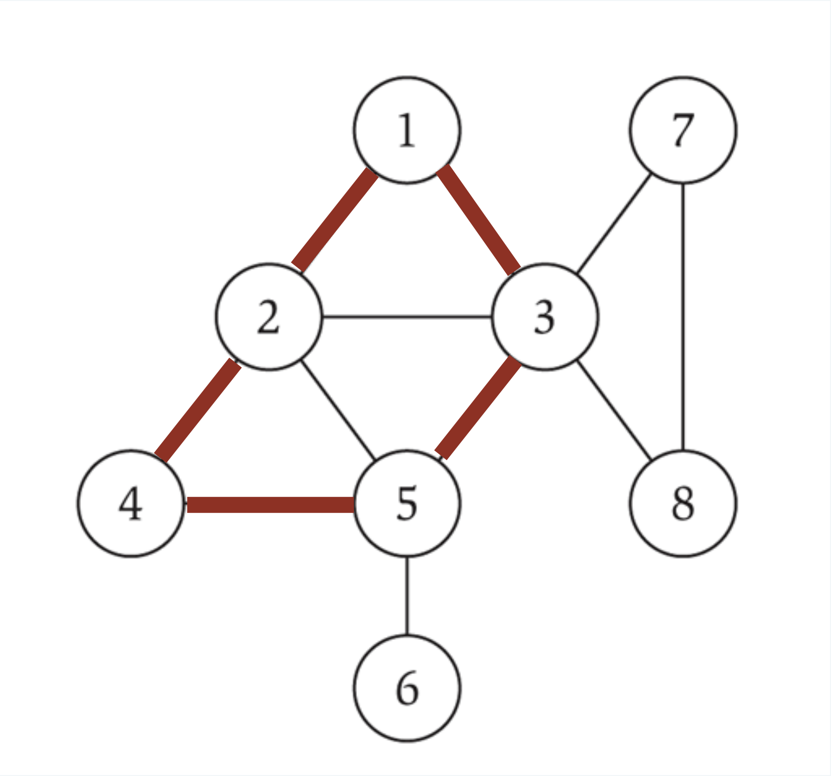
Def. Un cammino è semplice se tutti i suoi nodi sono distinti.

Def. Un grafo non orientato è connesso se per ogni coppia di nodi u e v, esiste un cammino tra u e v.



Cicli

Def. Un ciclo è un cammino $v_1, v_2, \ldots, v_k$ in cui $v_1 = v_k$, $k > 3$, e i primi k-1 nodi sono tutti distinti.



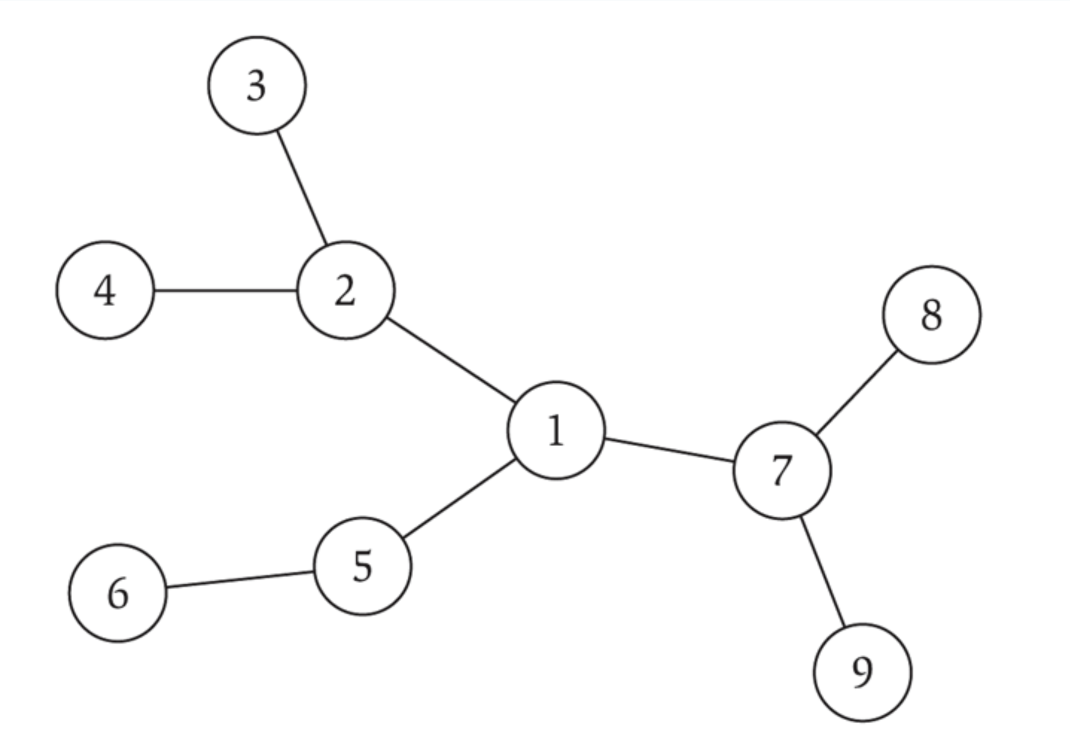
ciclo C = 1–2–4–5–3–1

Alberi

Def. Un grafo non orientato è un albero se è connesso e non contiene cicli.

Teorema. Sia G un grafo non orientato su n nodi. Due qualunque dei seguenti enunciati implicano il terzo:

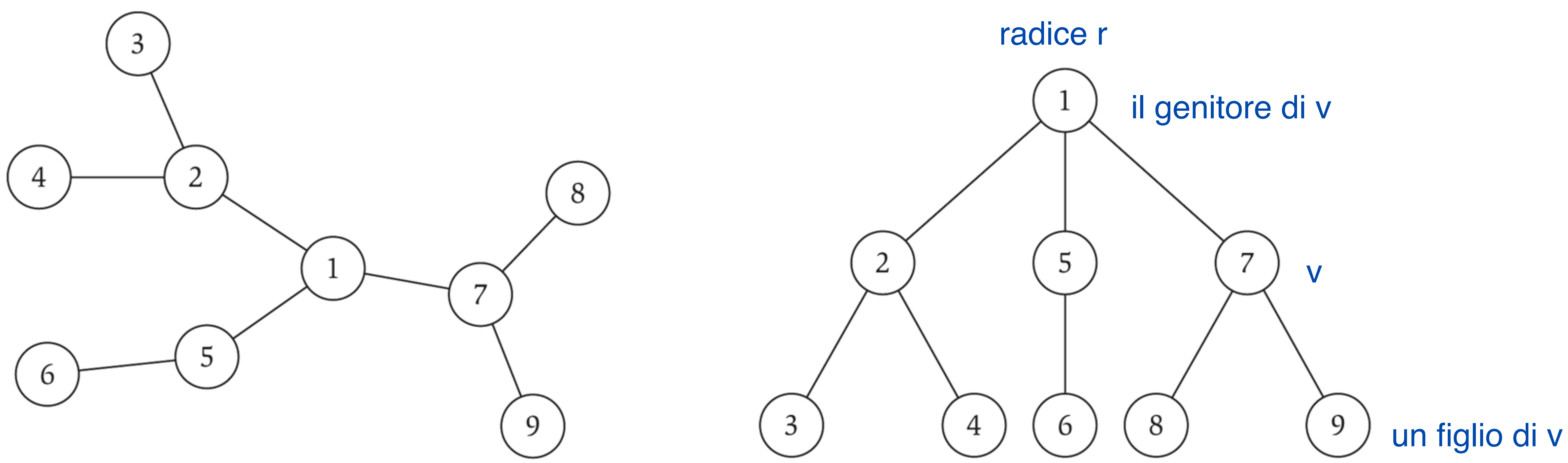
- G è connesso.
- G non contiene cicli.
- G ha $n - 1$ archi.



Alberi radicati

Dato un albero T, scegliamo un nodo radice r e orientiamo ciascun arco nel verso opposto ad r.

Importanza. Consente di modellare strutture gerarchiche.

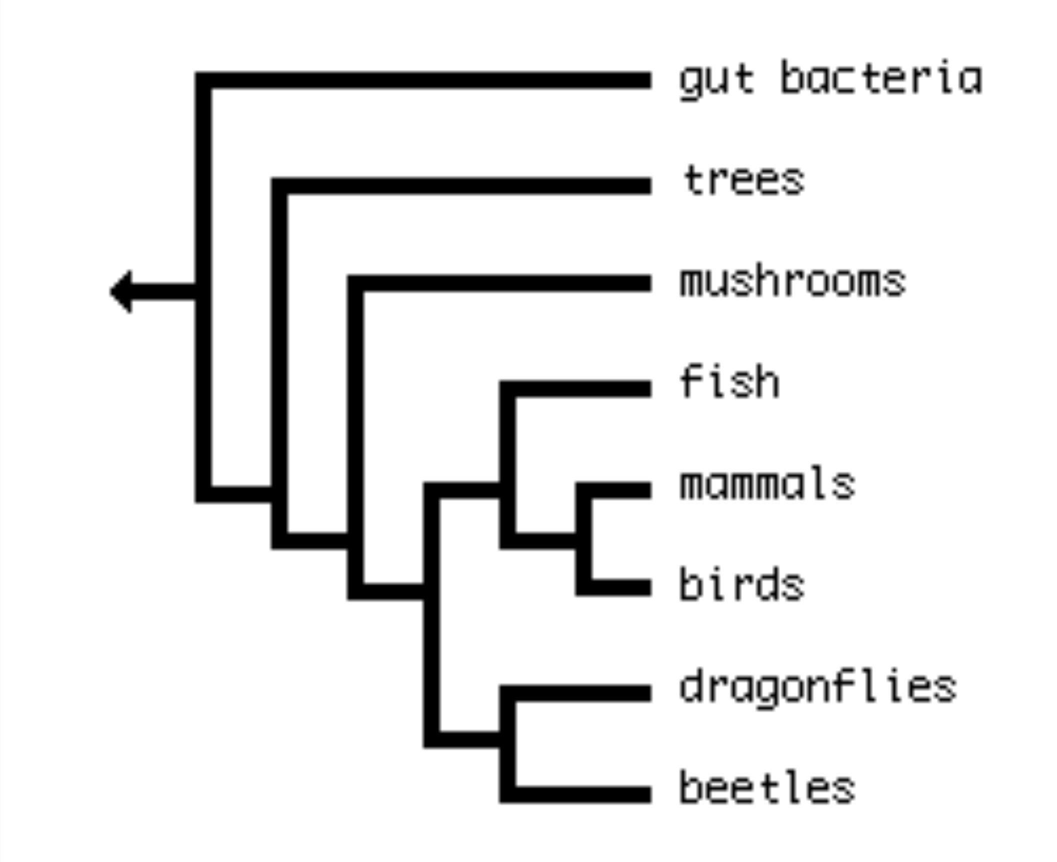


un albero

lo stesso albero, radicato su 1

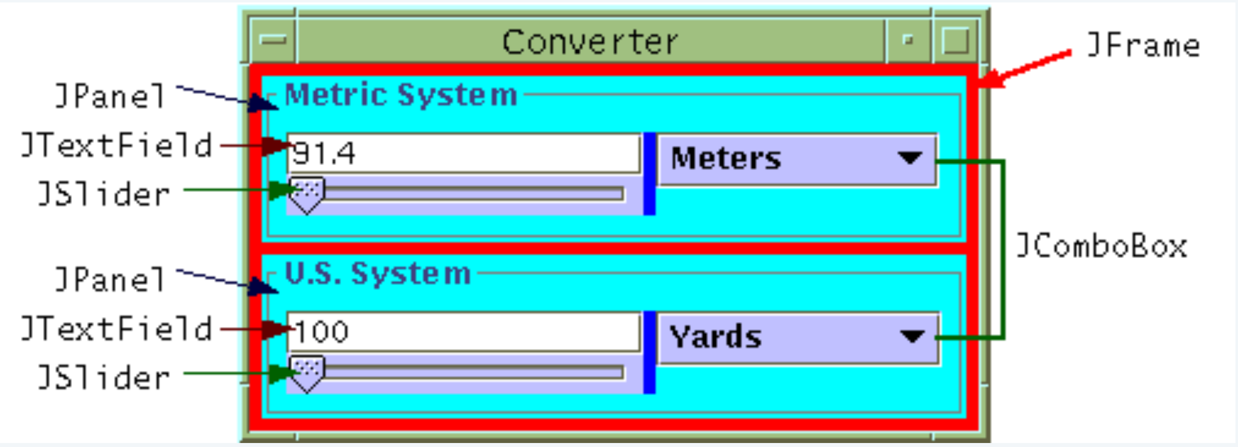
Alberi filogenetici

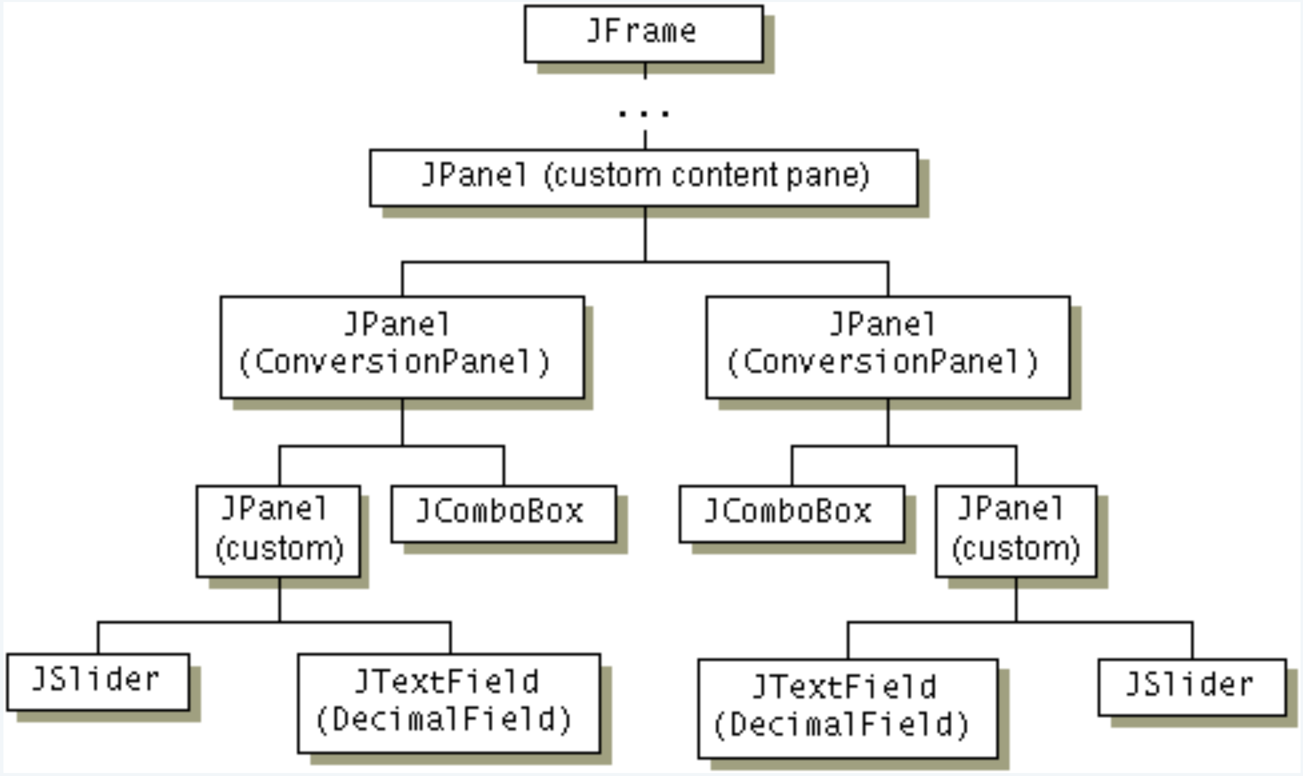
Descrivono la storia evolutiva delle specie.



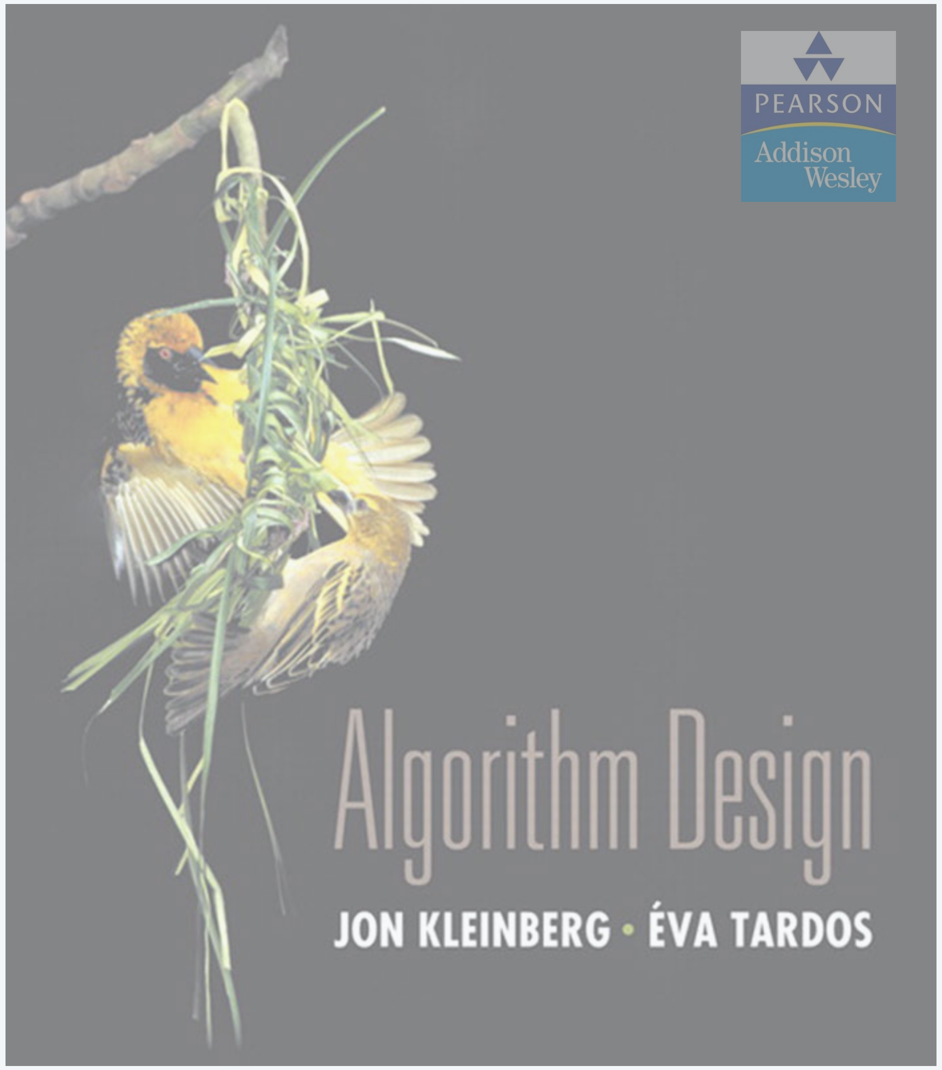
Gerarchia di contenimento di interfacce grafiche utente (GUI)

Descrive l'organizzazione dei componenti di una GUI.





http://java.sun.com/docs/books/tutorial/uiswing/overview/anatomy.html



3. Grafi

- *basic definitions and applications*
- *connettività e visite di grafi*
- *testing bipartiteness*
- *connectivity in directed graphs*
- *DAGs and topological ordering*

Connettività

Problema della connettività s-t. Dati due nodi s e t, esiste un cammino da s a t ?

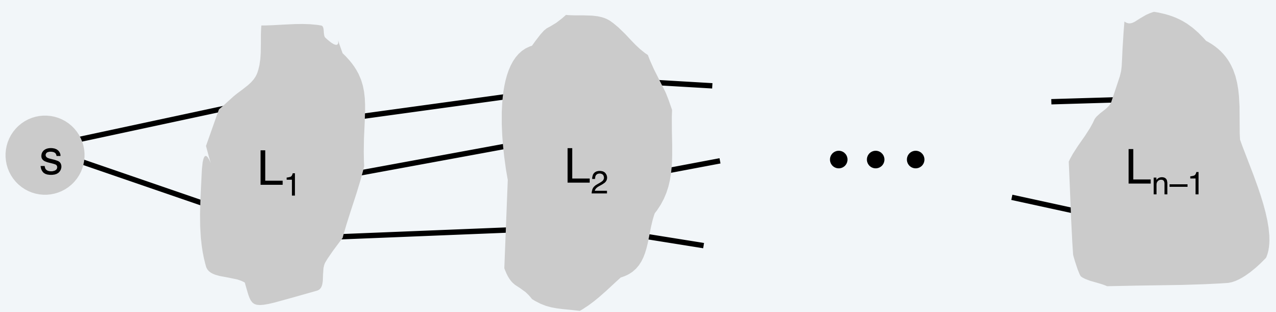
Problema del cammino minimo s-t. Dati due nodi s e t, qual è la lunghezza di un cammino minimo da s a t ?

Applicazioni.

- Facebook.
- Visita di labirinti.
- Numero di Kevin Bacon.
- Numero di inoltri minimi in una rete di comunicazione.

Visita in ampiezza (BFS - Breadth-First Search)

Intuizione. Esplorare da s in tutte le possibili direzioni, aggiungendo nodi uno "strato" alla volta.



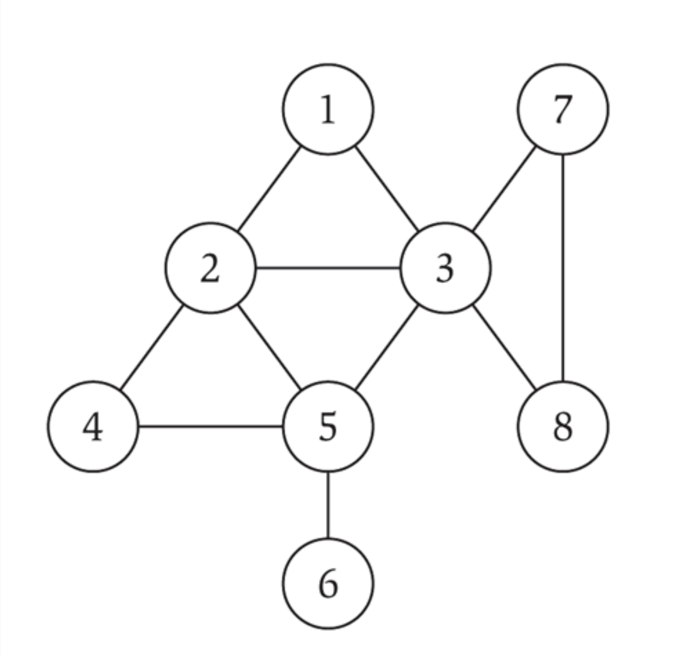
Algoritmo BFS.

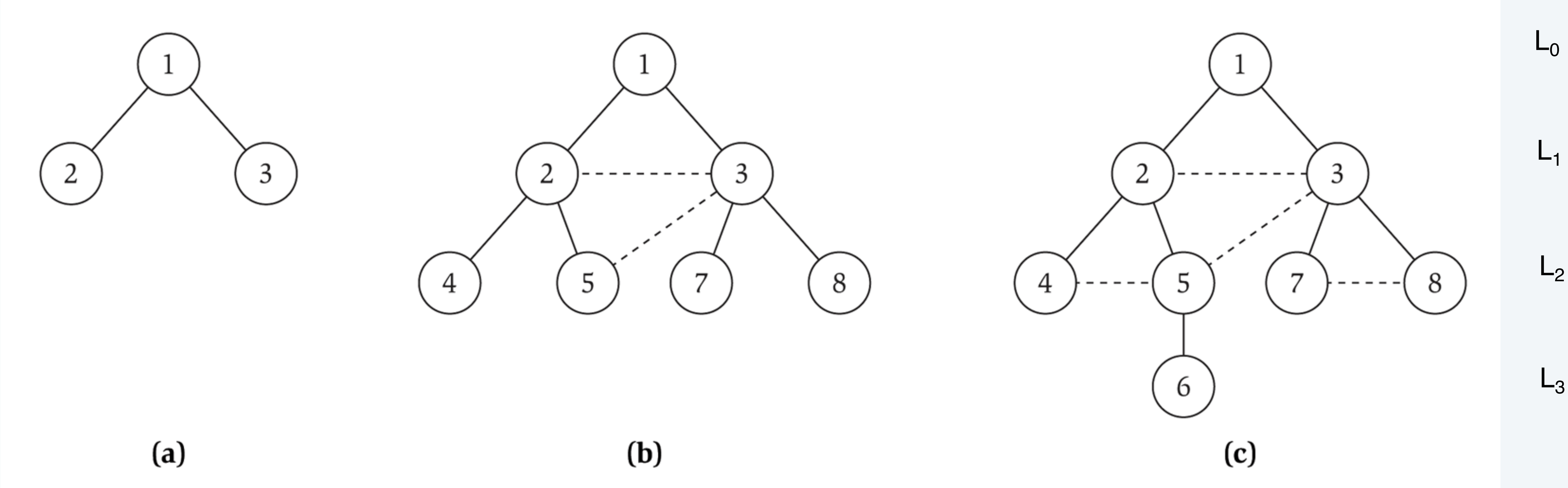
- $L_0 = \{ s \}$.
- L_1 = tutti i vicini di L_0.
- L_2 = tutti i nodi non contenuti in L_0 o L_1, e che hanno un arco da un nodo in L_1.
- L_{i+1} = tutti i nodi non contenuti in un livello precedente, e che hanno un arco da un nodo in L_i.

Teorema. Per ogni i, L_i consiste di tutti i nodi a distanza esattamente i da s. Esiste un cammino da s a t sse t appare in qualche strato.

Visita in ampiezza

Proprietà. Sia T un albero BFS di $G = (V, E)$, e sia (x, y) un arco di G. Allora, i livelli di x e di y differiscono tra loro di al più 1.





Visita in ampiezza: analisi

Teorema. L'implementazione vista sopra della BFS usa tempo $O(m + n)$ se il grafo è dato nella sua rappresentazione in liste di adiacenza.

Dim.

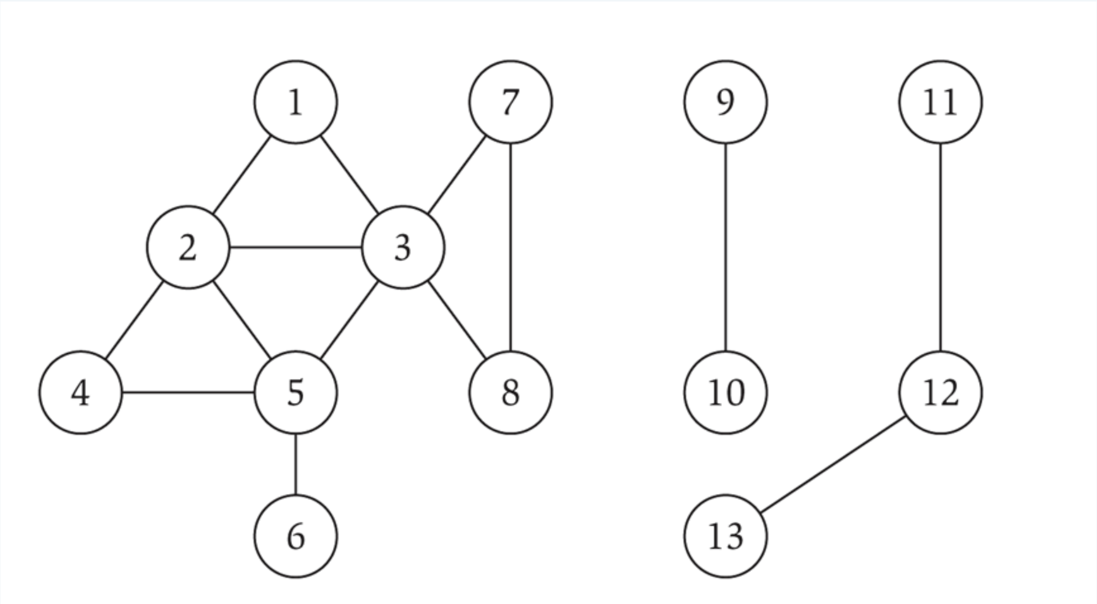
- Facile dimostrare tempo di esecuzione $O(n^2)$:
  - al più n liste $L[i]$
  - ogni nodo occorre in al più una lista; il ciclo for esegue $\leq n$ volte
  - quando consideriamo il nodo u, ci sono $\leq n$ archi incidenti (u, v), e spendiamo $O(1)$ per processare ciascun arco

- In effetti il tempo di esecuzione è $O(m + n)$:
  - sul nodo u, ci sono $grado(u)$ archi incidenti (u, v)
  - il tempo totale di processamento degli archi è $\Sigma_{u \in V}\ grado(u) = 2m$. ▪

ogni arco (u, v) è contato due volte
nella somma: una in grado(u) ed una in grado(v)

Componenti connesse

Componente connessa. Trovare tutti i nodi raggiungibili da s.



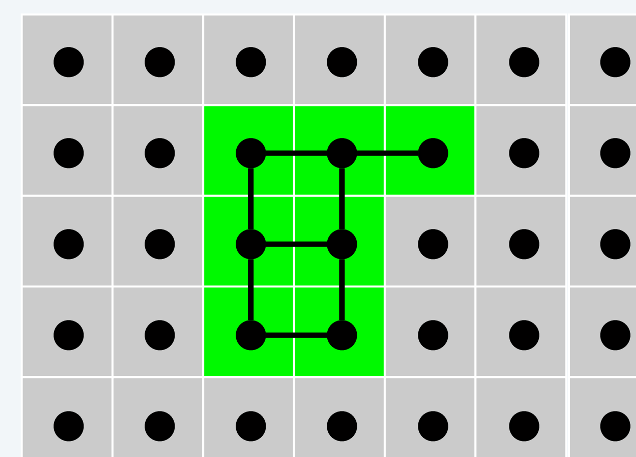
Componente connessa contenente il nodo $1 = \{ 1, 2, 3, 4, 5, 6, 7, 8 \}$.

Flood fill [Riempimento ad allagamento]

Flood fill. Dato un pixel verde in un immagine, cambiare in blu il colore di un intero blob di pixel verdi collegati.

- Nodo: pixel.
- Arco: due pixel verdi adiacenti.
- Blob: componente connessa di pixel verdi.

ricolora il blob verde di blu

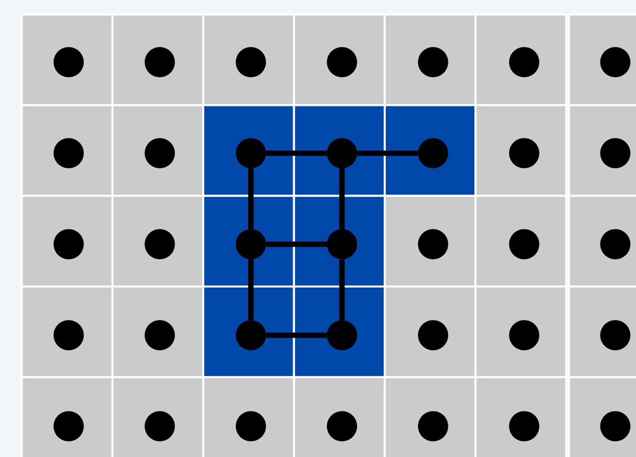


Flood fill [Riempimento ad allagamento]

Flood fill. Dato un pixel verde in un immagine, cambiare in blu il colore di un intero blob di pixel verdi collegati.

- Nodo: pixel.
- Arco: due pixel verdi adiacenti.
- Blob: componente connessa di pixel verdi.

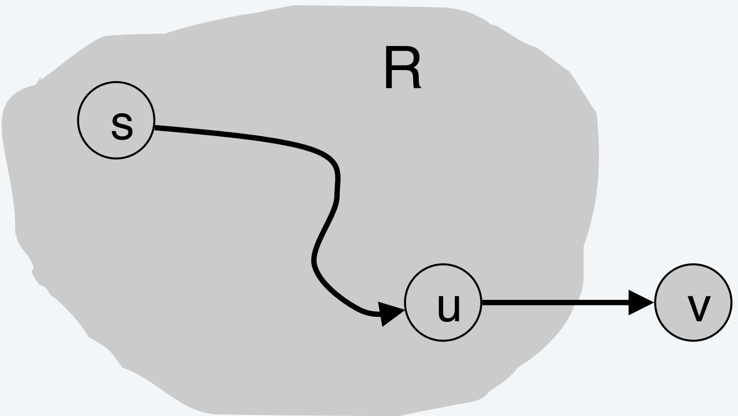
ricolora il blob verde di blu



Flood fill [Riempimento ad allagamento]

Componente connessa. Trovare tutti i nodi raggiungibili da s.

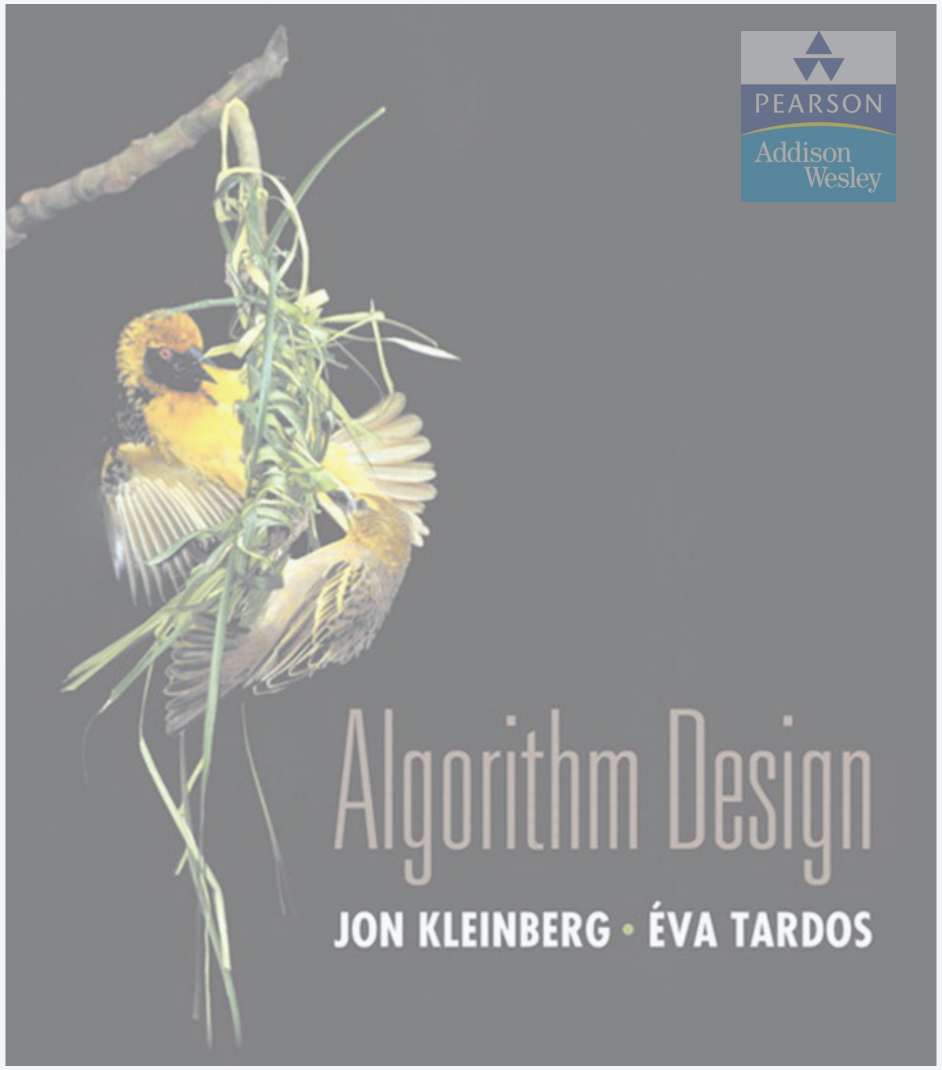
```
R will consist of nodes to which s has a path
Initially R = {s}
While there is an edge (u, v) where u ∈ R and v ∉ R
  Add v to R
Endwhile
```



v può essere senz'altro aggiunto

Teorema. Al termine, R è la componente connessa contenente s.

- BFS = esplora in ordine di distanza da s.
- DFS = esplora in ordine diverso (dà priorità all'esplorazione in profondità).



3. Grafi

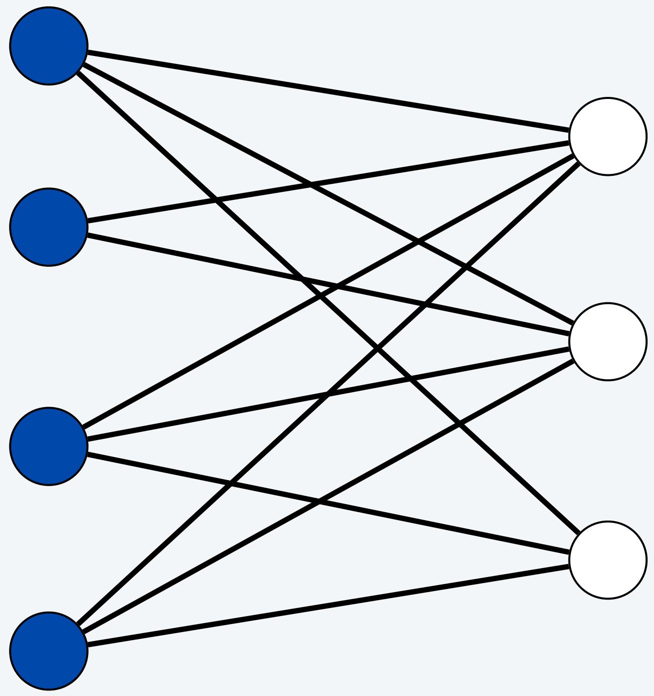
- *basic definitions and applications*
- *graph connectivity and graph traversal*
- *grafi bipartiti*
- *connectivity in directed graphs*
- *DAGs and topological ordering*

Grafi bipartiti

Def. Un grafo non orientato $G = (V, E)$ è bipartito se i nodi possono essere colorati di blu o di bianco in modo che ogni arco abbia un estremo blu ed uno bianco.

Applicazioni.

- Abbinamenti: specializzandi = blu, ospedali = bianco.
- Schedulazione: macchine = blu, lavori = bianco.



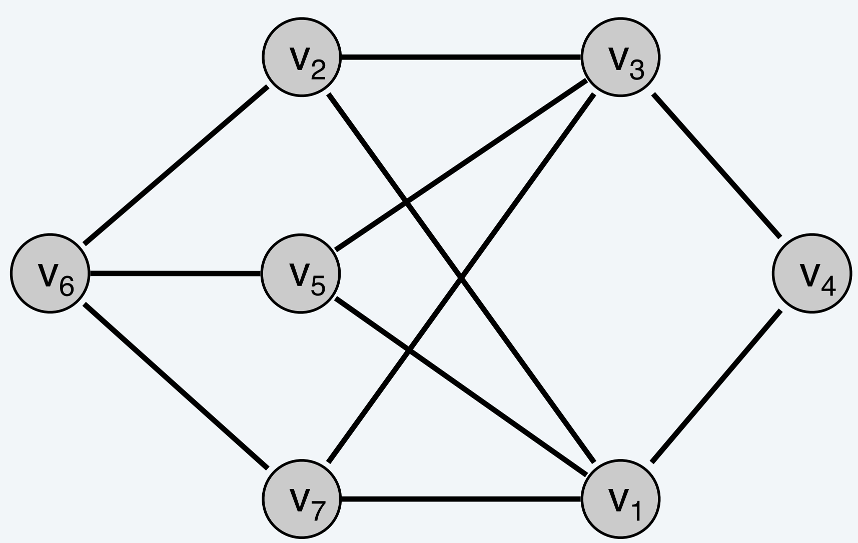
un grafo bipartito

Verifica di bipartibilità

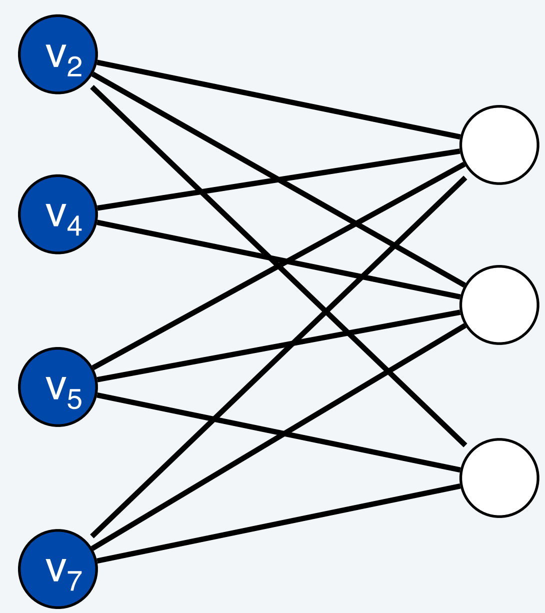
Molti problemi su grafi divengono:

- Più semplici se il grafo considerato è bipartito (es.: abbinamento).
- Trattabili se il grafo considerato è bipartito (es.: insieme indipendente).

Prima di tentare il progetto di un algoritmo, dobbiamo comprendere la struttura dei grafi bipartiti.



un grafo bipartito G

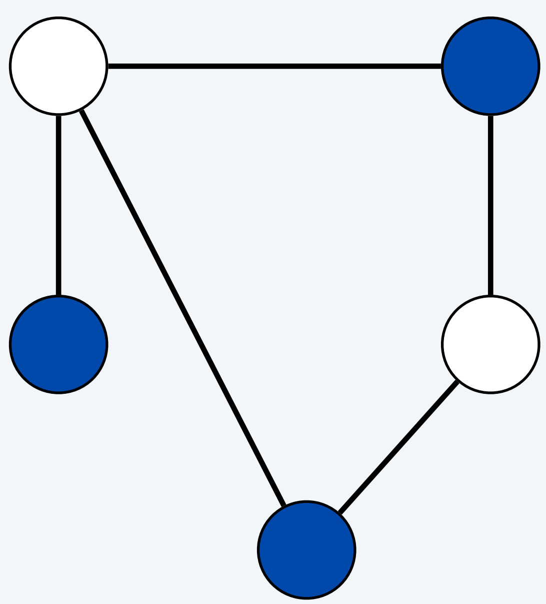


un altro disegno di G

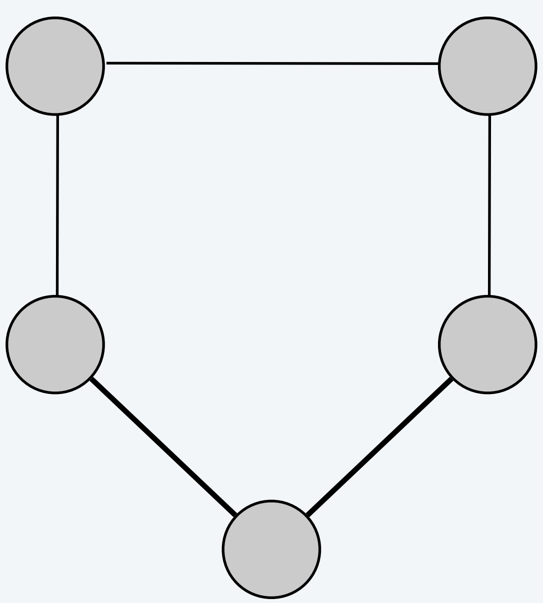
Un ostacolo alla bipartibilità

Lemma. Se un grafo G è bipartito, non può contenere un ciclo di lunghezza dispari.

Dim. Non è possibile 2-colorare il ciclo di lunghezza dispari; tanto meno G.



bipartito
(2-colorabile)



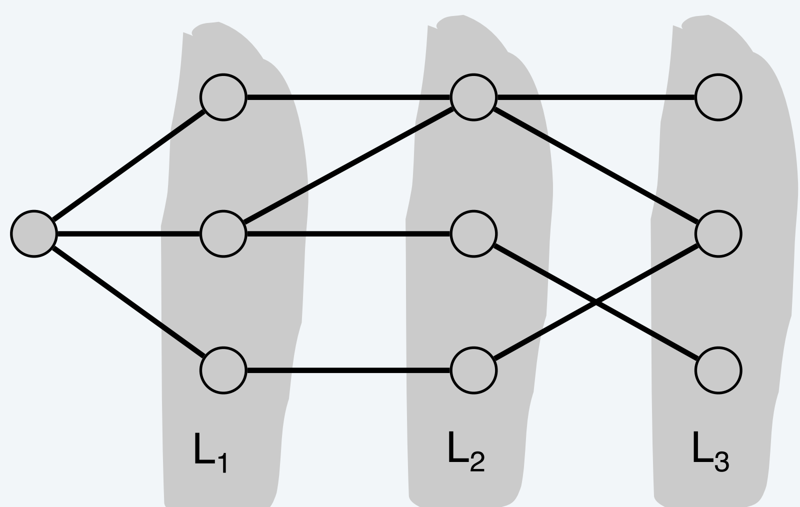
non bipartito
(non 2-colorabile)

Grafi bipartiti

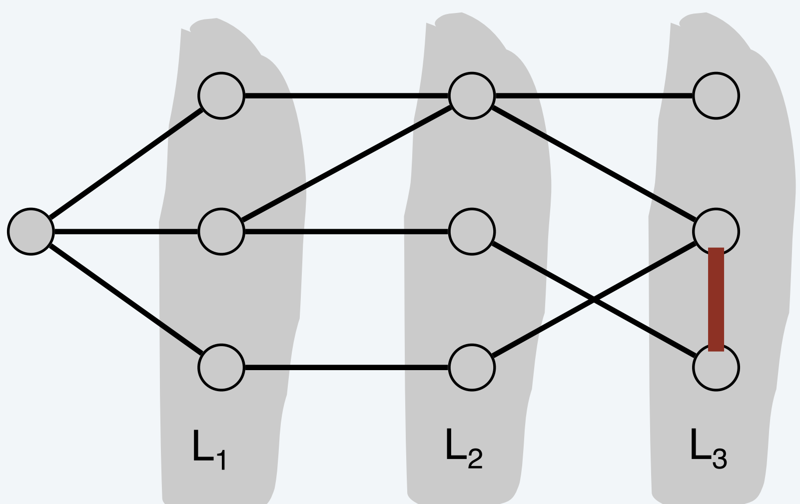
Lemma. Sia G un grafo connesso, e siano $L_0, \ldots, L_k$ gli strati di una BFS originata al nodo s. Vale esattamente una delle seguenti.

(i) Nessun arco di G collega due nodi dello stesso strato, e G è bipartito.

(ii) Qualche arco di G collega due nodi dello stesso strato, e G contiene un ciclo di lunghezza dispari (e quindi non è bipartito).



Caso (i)



Caso (ii)

Grafi bipartiti

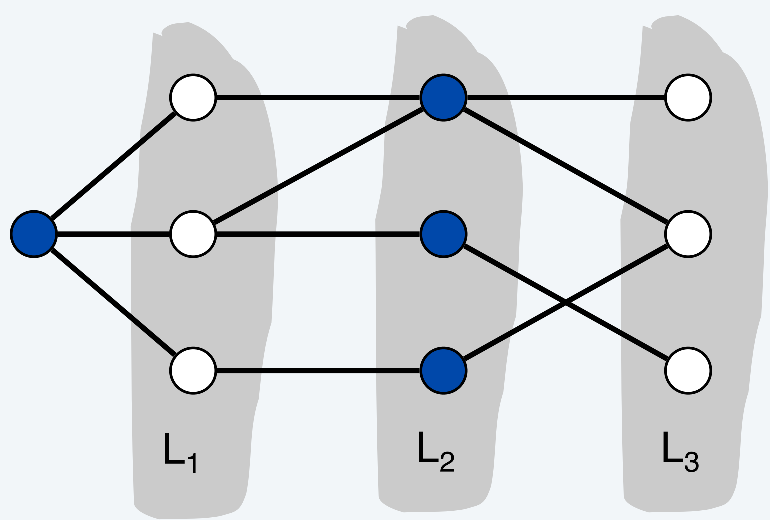
Lemma. Sia G un grafo connesso, e siano $L_0, \ldots, L_k$ gli strati di una BFS originata al nodo s. Vale esattamente una delle seguenti.

(i) Nessun arco di G collega due nodi dello stesso strato, e G è bipartito.

(ii) Qualche arco di G collega due nodi dello stesso strato, e G contiene un ciclo di lunghezza dispari (e quindi non è bipartito).

Dim. (i)

- Supponiamo che nessun arco colleghi due nodi dello stesso strato.
- Per le proprietà della BFS, ogni arco collega nodi di livelli adiacenti.
- Bipartizione: bianco = nodi su livelli dispari, blu = nodi su livelli pari.



Case (i)

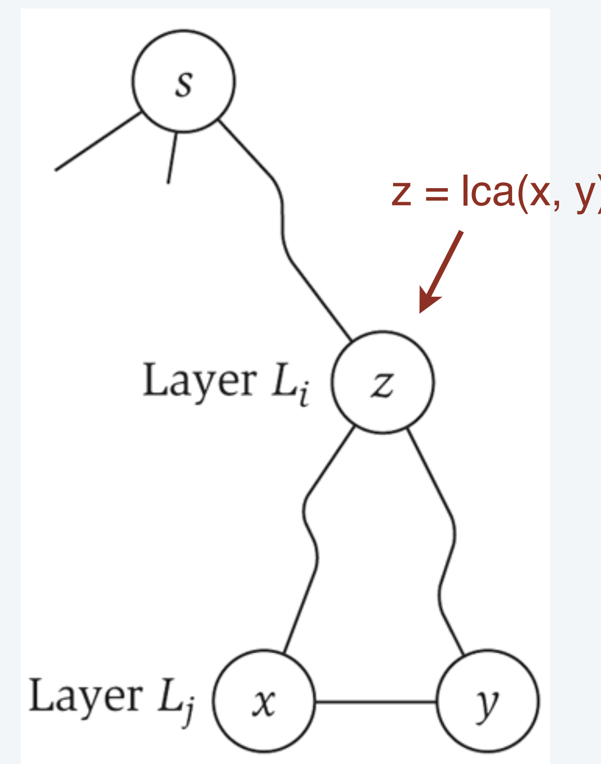
Lemma. Sia G un grafo connesso, e siano $L_0, \ldots, L_k$ gli strati di una BFS originata al nodo s. Vale esattamente una delle seguenti.

(i) Nessun arco di G collega due nodi dello stesso strato, e G è bipartito.

(ii) Qualche arco di G collega due nodi dello stesso strato, e G contiene un ciclo di lunghezza dispari (e quindi non è bipartito).

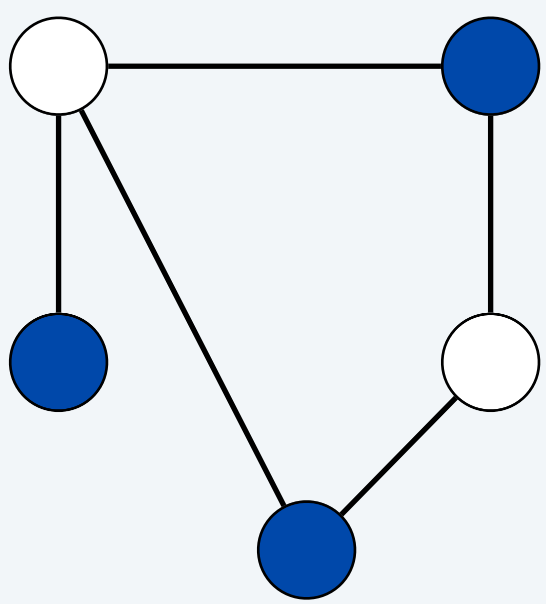
Dim. (ii)

- Sia (x, y) un arco con x, y nello stesso livello L_j.
- Sia $z = lca(x, y)$ = l'antenato comune di livello infimo.
- Sia L_i il livello contenente z.
- Consideriamo il ciclo composto dall'arco da x ad y, poi dal cammino da y a z, poi dal cammino da z ad x.
- La lunghezza è $\underbrace{1}_{\text{(x, y)}} + \underbrace{(j-i)}_{\text{path from y to z}} + \underbrace{(j-i)}_{\text{path from z to x}}$, che è dispari. ▪

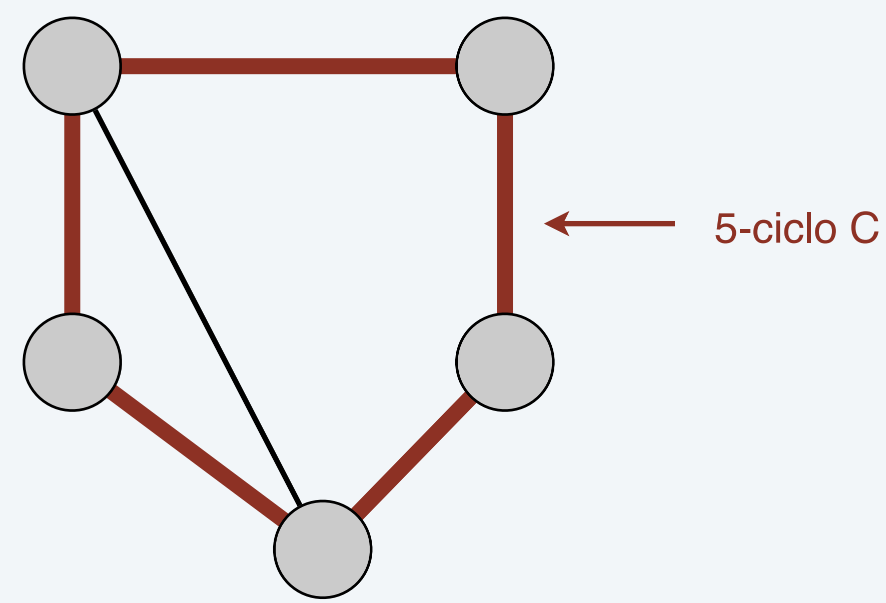


L'unico ostacolo alla bipartibilità

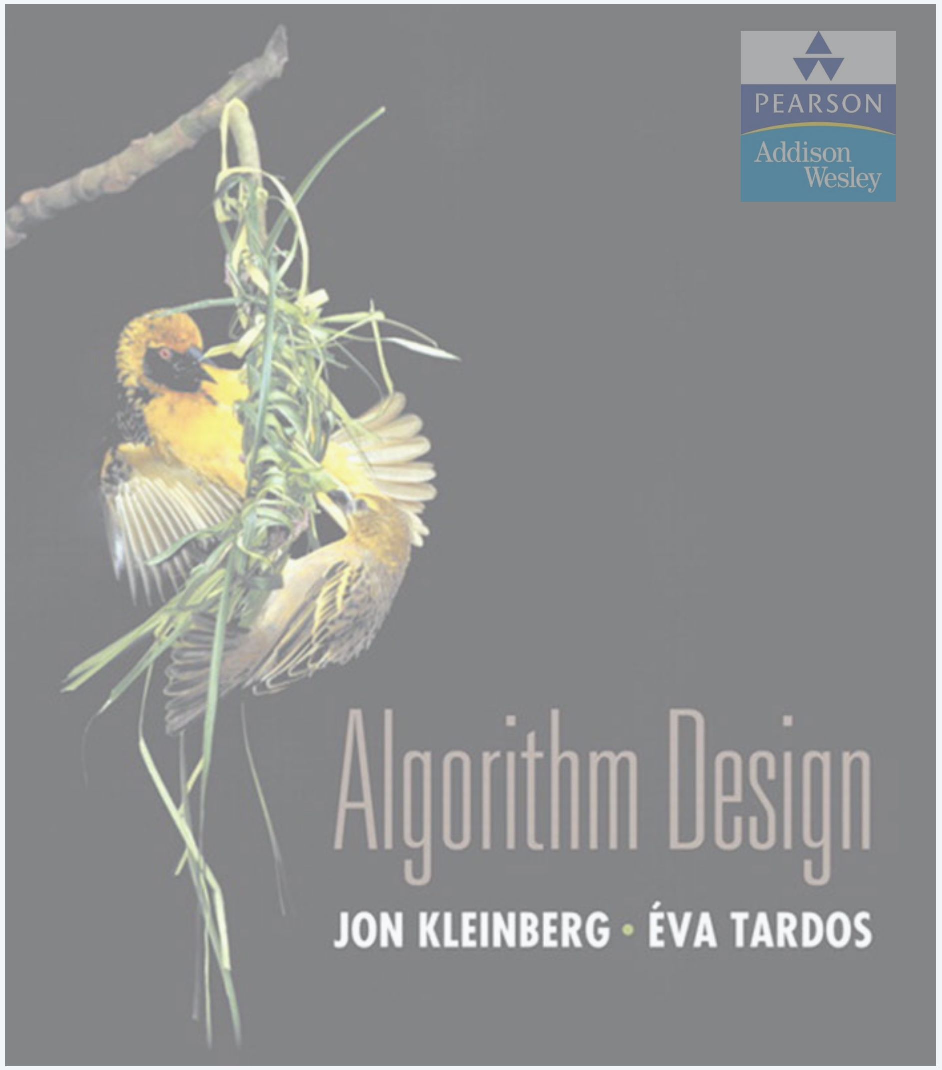
Corollario. Un grafo G è bipartito sse non contiene cicli di lunghezza dispari.



bipartito
(2-colorabile)



non bipartito
(non 2-colorabile)



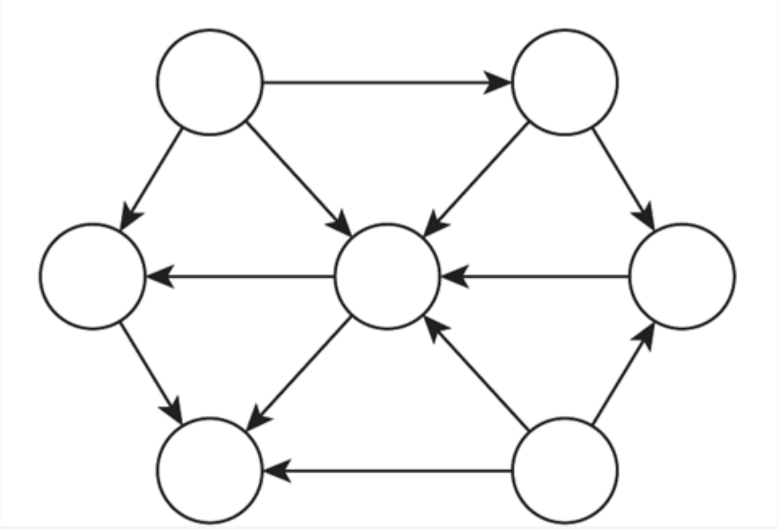
3. GRAFI

- *basic definitions and applications*
- *graph connectivity and graph traversal*
- *testing bipartiteness*
- *connettività nei digrafi*
- *DAGs and topological ordering*

Digrafi [directed graphs]

Notazione. $G = (V, E)$.

- Arco (u, v) parte dal nodo u ed arriva nel nodo v.



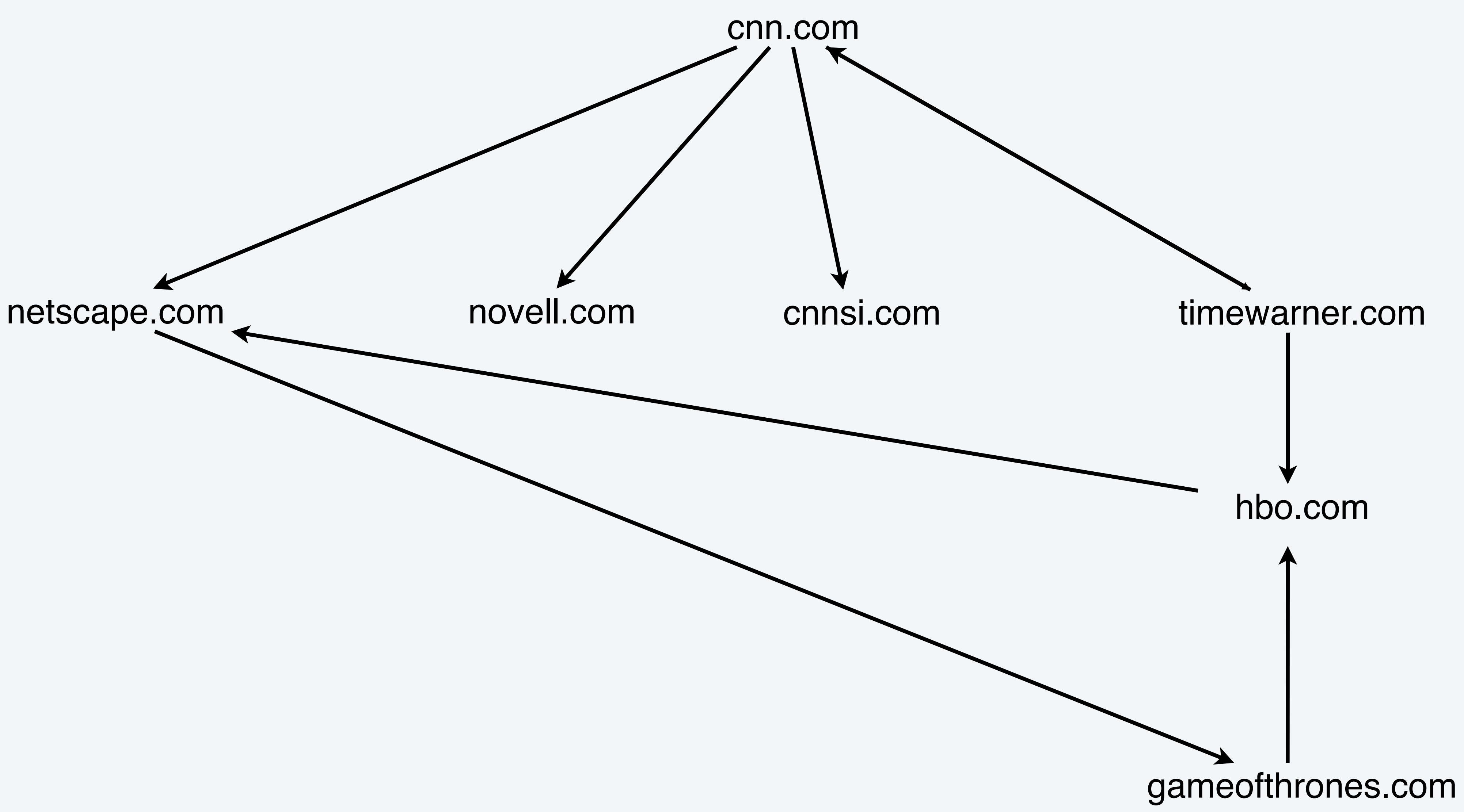
Es. Grafo del web: un hyperlink punta da una pagina web ad un'altra.

- Il verso degli archi è cruciale.
- I moderni motori di ricerca web sfruttano la struttura degli hyperlink per ordinare le pagine web per importanza e autorevolezza.

World wide web

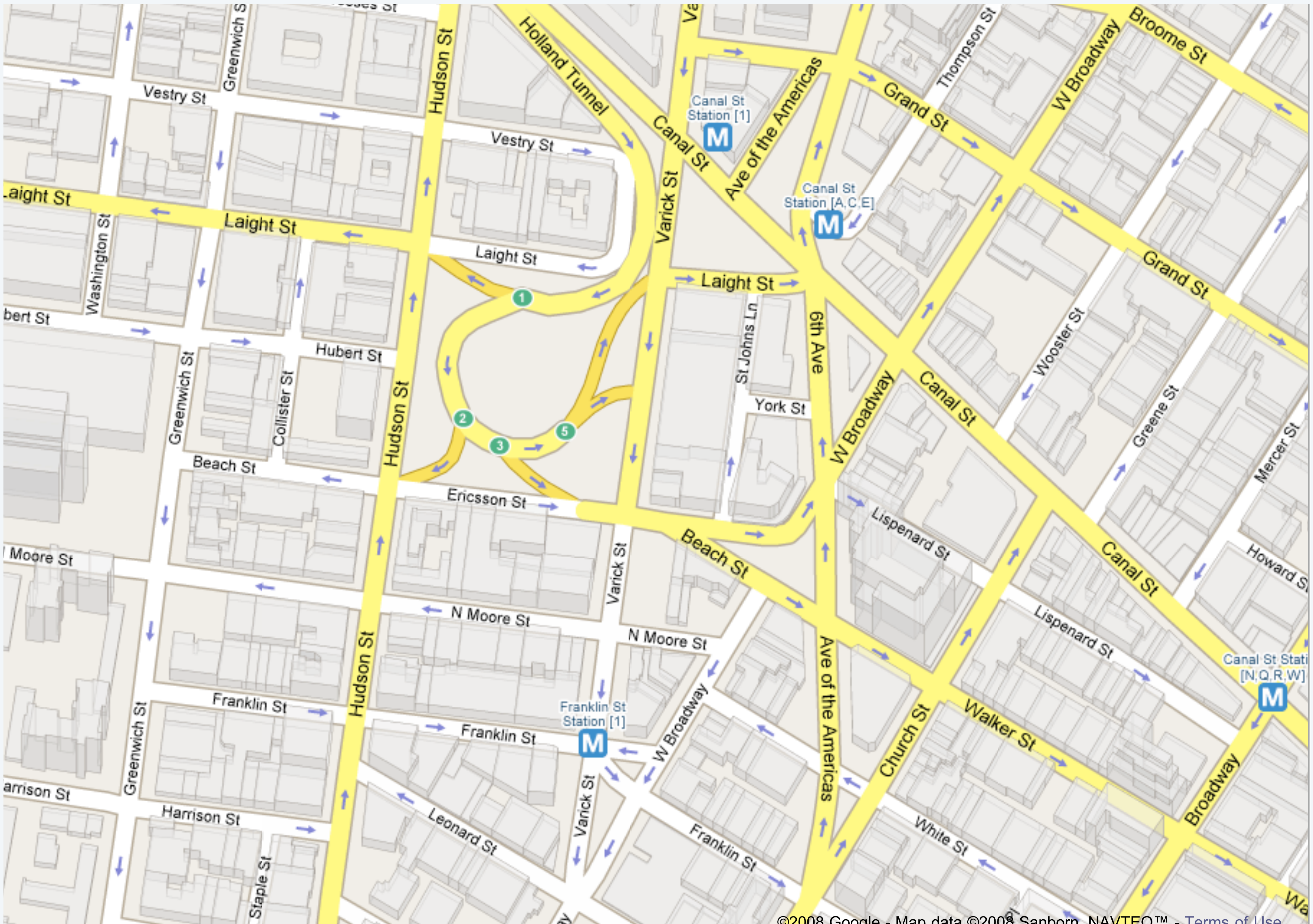
Grafo del web.

- Nodo: pagina web.
- Arco: hyperlink da una pagina ad un'altra (il verso è cruciale).
- I moderni motori di ricerca sfruttano la struttura degli hyperlink per ordinare le pagine web per importanza.



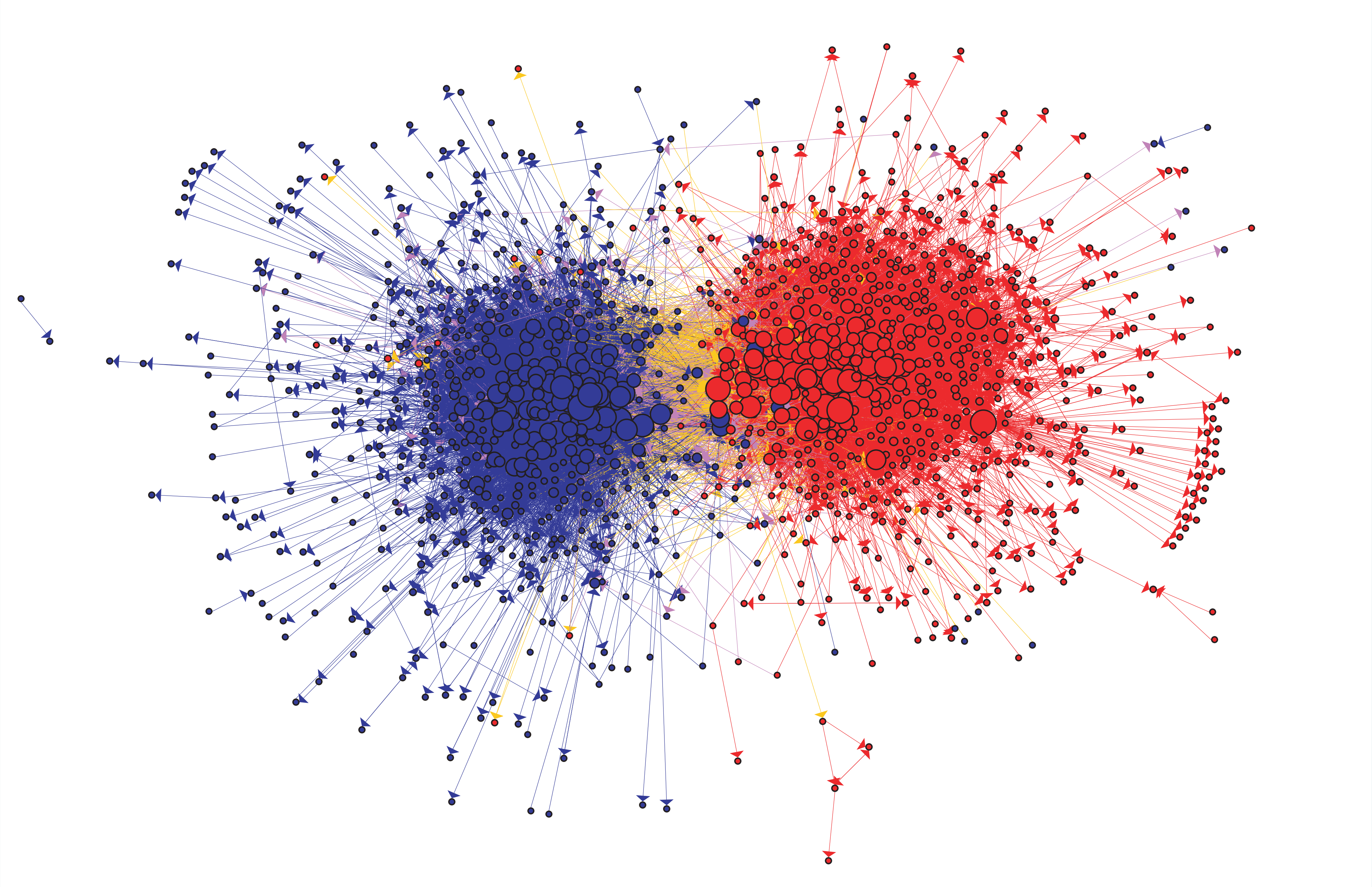
Rete stradale

Nodo = intersezione; arco = via a senso unico.



Grafo della blogosfera politica

Nodo = blog di politica; arco = link.

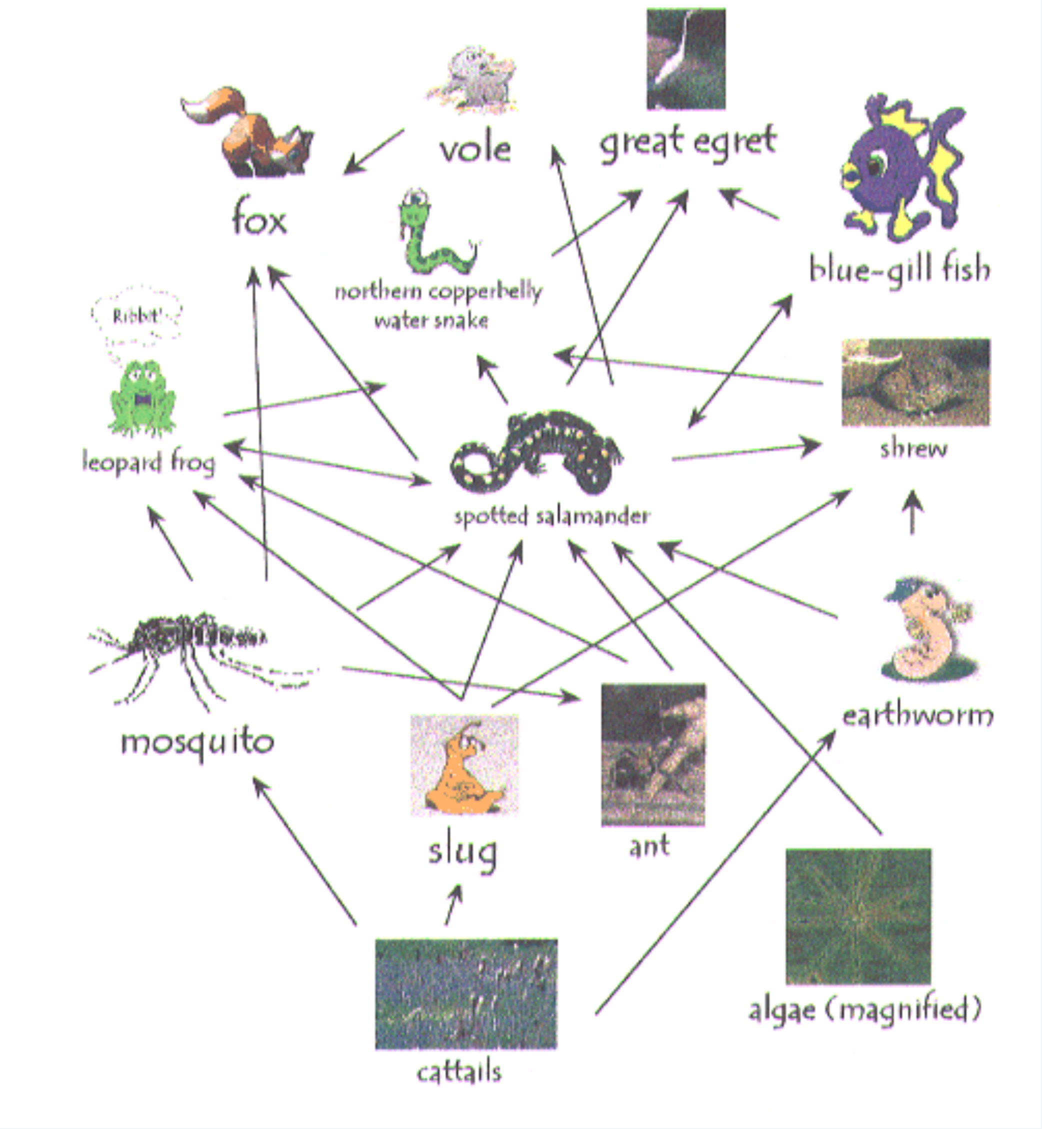


The Political Blogosphere and the 2004 U.S. Election: Divided They Blog, Adamic and Glance, 2005

Rete alimentare di un ecosistema

Rete alimentare.

- Nodo = specie.
- Arco = da preda a predatore.



Riferimento: http://www.twingroves.district96.k12.il.us/Wetlands/Salamander/SalGraphics/salfoodweb.giff

Alcune applicazioni dei digrafi

digrafo	nodo	arco
di trasporto	intersezione stradale	via a senso unico
del web	pagina web	hyperlink
rete alimentare	specie	relazione preda-predatore
WordNet	insieme di sinonimi	iperonimia
schedulazione	lavoro	vincolo di precedenza
finanziario	banca	transazione
telefono cellulare	persona	chiamata effettuata
malattia infettiva	persona	infezione
gioco	situazione sul tabellone	mossa lecita
delle citazioni	articolo di rivista	citazione
object graph	object	pointer
gerarchia di classi	classe	eredita da
flusso di controllo	blocco di codice	salto

Visita di grafi

Raggiungibilità. Dato un nodo s, trovare tutti i nodi raggiungibili da s.

Problema dei cammini minimi s↝t. Dati due nodi s e t, qual è la lunghezza di un cammino minimo da s a t ?

Visita di digrafi. La BFS si estende in modo naturale ai digrafi.

Crawler Web. Partire dalla pagina web s. Trovare tutte le pagine web raggiungibili a partire da s, in modo diretto o meno.

Connettività forte

Def. I nodi u e v sono mutuamente raggiungibili se c'è sia una cammino da u a v che un cammino da v a u.

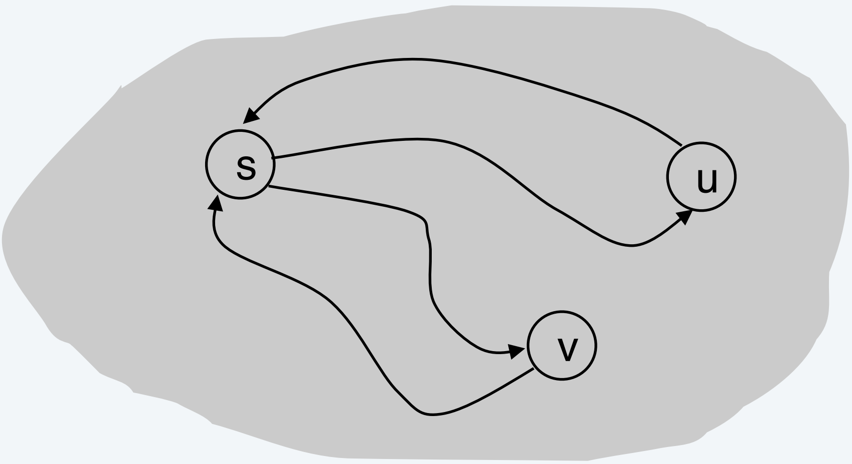
Def. Un grafo è fortemente connesso se ogni coppia di nodi è mutuamente raggiungibile.

Lemma. Sia s un nodo qualunque. G è fortemente connesso sse ogni nodo è raggiungibile da s, e s è raggiungibile da ogni nodo.

Dim. $\Rightarrow$ Segue dalla definizione.

Dim. $\Leftarrow$ Cammino da u a v: concatenare il cammino $u \rightsquigarrow s$ col cammino $s \rightsquigarrow v$.

Cammino da v a u: concatenare il cammino $v \rightsquigarrow s$ col cammino $s \rightsquigarrow u$. ▪



i cammini possono sovrapporsi

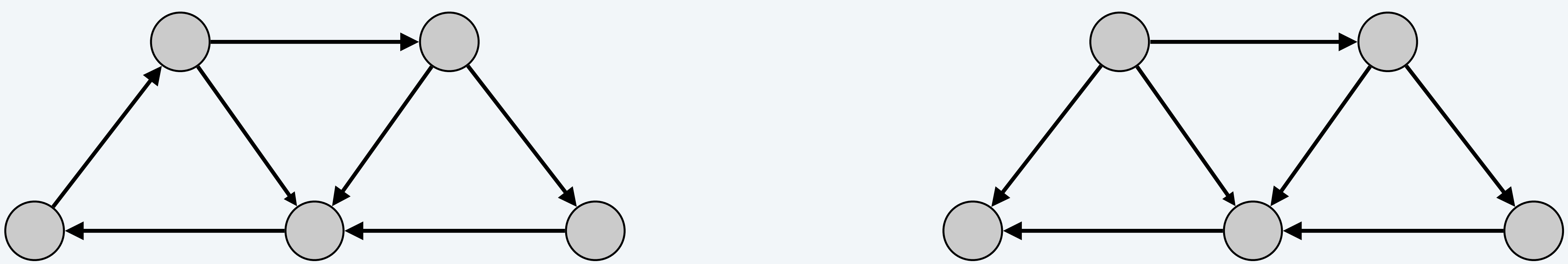
Connettività forte: algoritmo

Teorema. Si può determinare se G è fortemente connesso in tempo $O(m + n)$.

Dim.

- Selezioniamo un nodo qualunque s.
- Eseguiamo una BFS da s in G.
- Eseguiamo una BFS da s in $G^{reverse}$.
- Restituiamo true sse entrambe le BFS raggiungono tutti i nodi.
- La correttezza segue direttamente dal lemma precedente. ▪

rovescia il verso di ogni arco in G

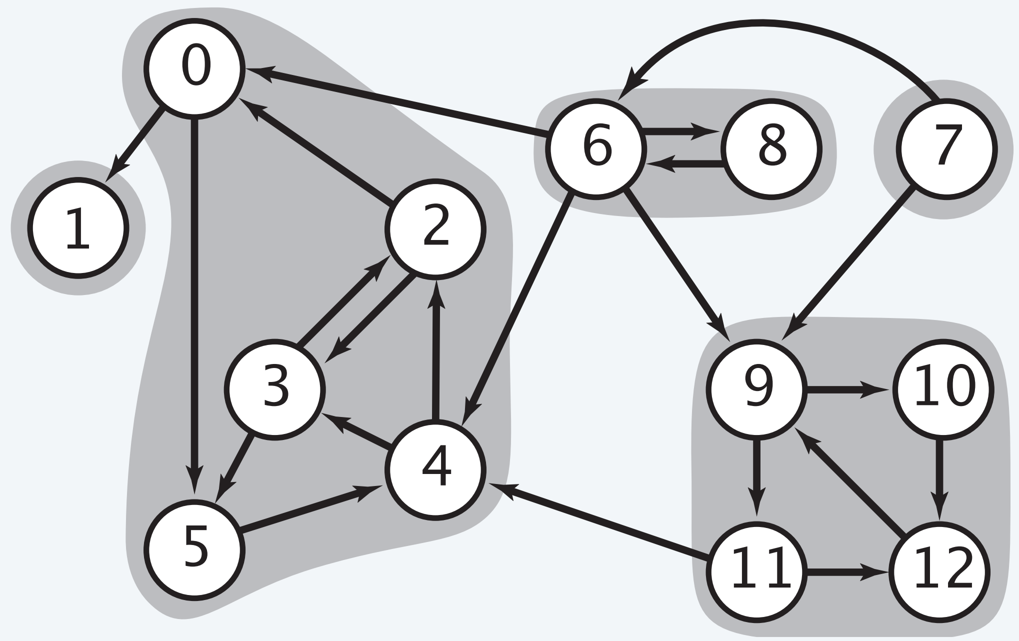


fortemente connesso

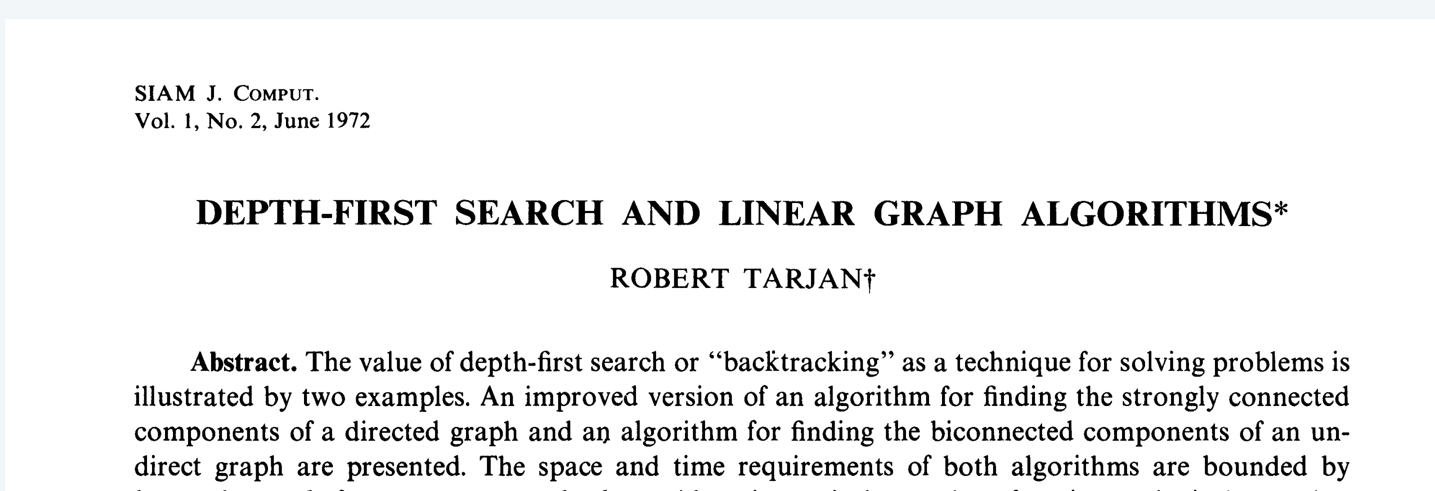
non fortemente connesso

Componenti fortemente connesse

Def. Una componente fortemente connessa è un sottoinsieme massimale di nodi mutuamente raggiungibili.



Teorema. [Tarjan 1972] Le componenti fortemente connesse possono essere determinate in tempo $O(m + n)$.

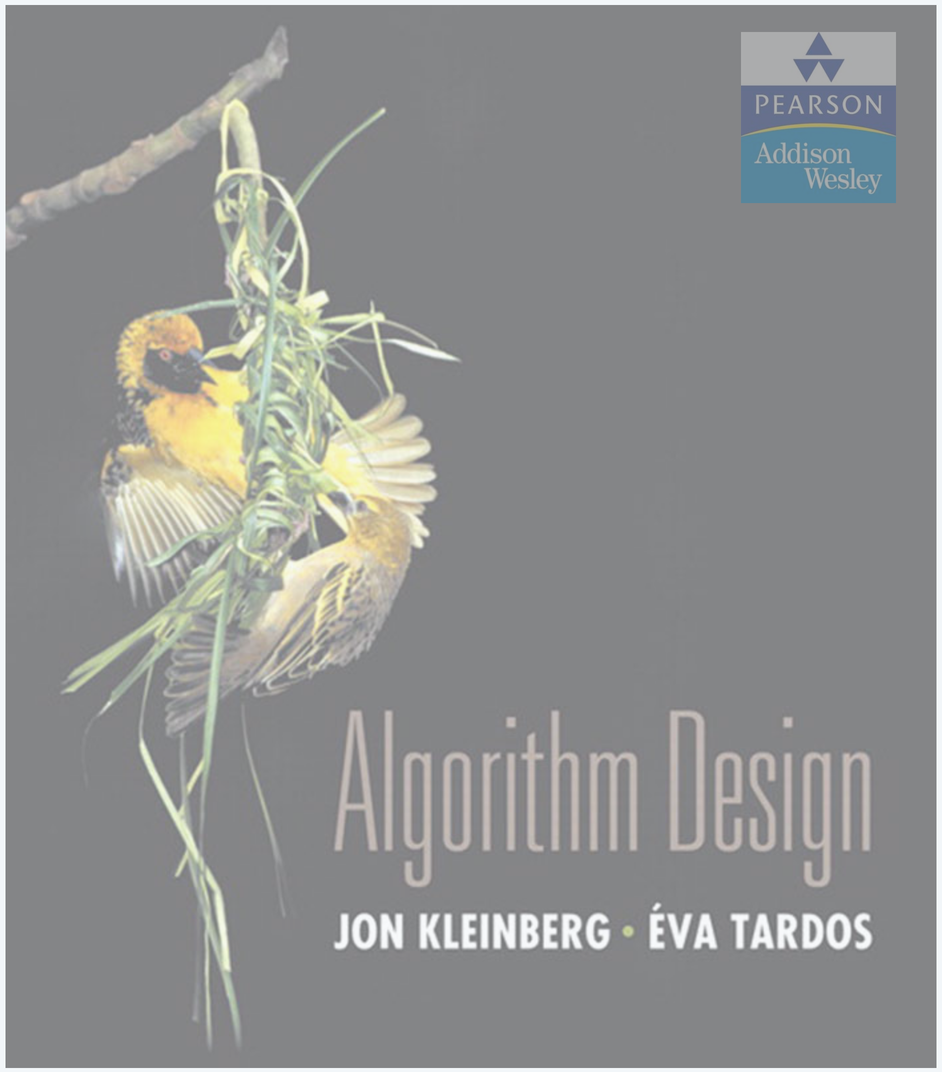


SIAM J. COMPUT.
Vol. 1, No. 2, June 1972

DEPTH-FIRST SEARCH AND LINEAR GRAPH ALGORITHMS*

ROBERT TARJAN†

Abstract. The value of depth-first search or "backtracking" as a technique for solving problems is illustrated by two examples. An improved version of an algorithm for finding the strongly connected components of a directed graph and an algorithm for finding the biconnected components of an undirect graph are presented. The space and time requirements of both algorithms are bounded by



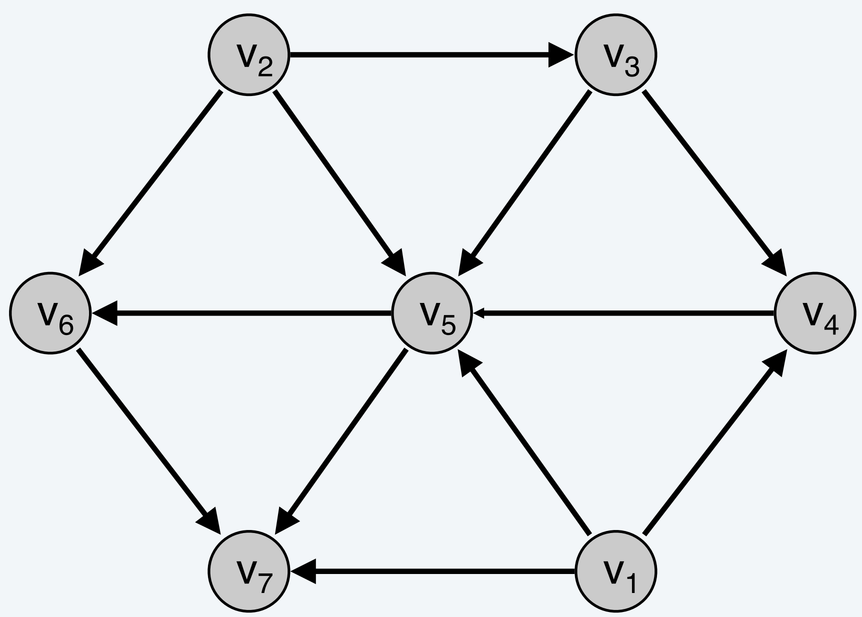
3. Grafi

- *basic definitions and applications*
- *graph connectivity and graph traversal*
- *testing bipartiteness*
- *connectivity in directed graphs*
- *DAG e ordinamento topologico*

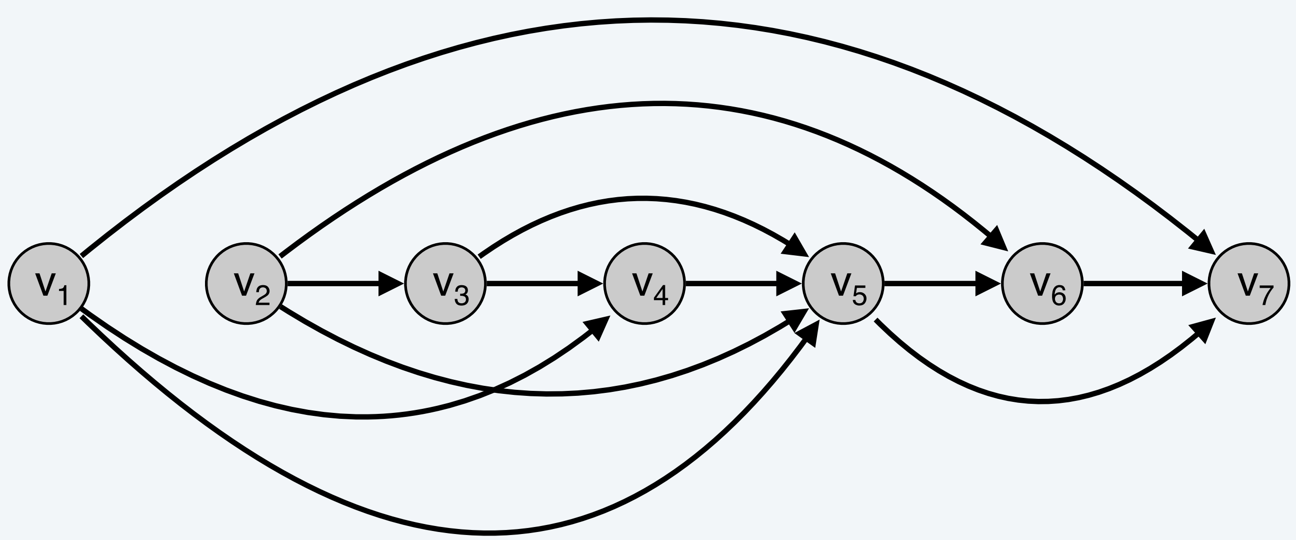
Digrafi aciclici [DAG]

Def. Un DAG è un digrafo che non contiene cicli orientati.

Def. Un ordinamento topologico di un digrafo $G = (V, E)$ è un ordinamento dei suoi nodi $v_1, v_2, \ldots, v_n$ tale che per ogni arco (v_i, v_j) abbiamo $i < j$.



un DAG



un ordinamento topologico

Vincoli di precedenza

Vincoli di precedenza. Un arco (v_i, v_j) denota che il compito v_i deve essere eseguito prima del compito v_j.

Applicazioni.

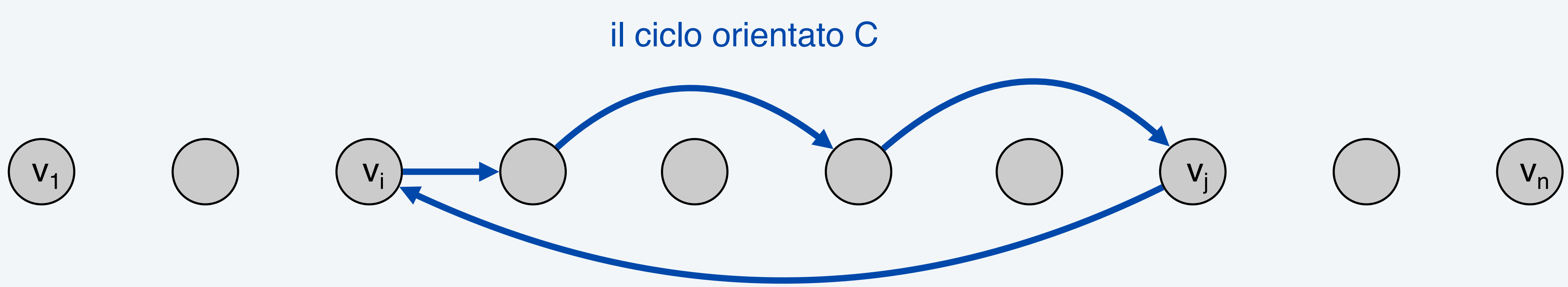
- Prerequisiti di insegnamenti: l'insegnamento v_i deve essere svolto prima di v_j.
- Compilazione: il modulo v_i deve essere compilato prima di v_j.
- *Pipeline* di processi computazionali: l'output del processo v_i è necessario a determinare l'input del processo v_j.

Digrafi aciclici

Lemma. Se G ha un ordinamento topologico, allora G è un DAG.

Dim. [per assurdo]

- Supponiamo G abbia un ordinamento topologico $v_1, v_2, \ldots, v_n$ e che G abbia anche un ciclo orientato C.
- Sia v_i il nodo di indice minimo in C, e sia v_j il nodo che precede v_i; quindi (v_j, v_i) è un arco.
- Per la nostra scelta di i, si ha $i < j$.
- D'altra parte, poiché (v_j, v_i) è un arco e $v_1, v_2, \ldots, v_n$ è un ordine topologico, dobbiamo avere $j < i$, contraddizione. ▪



l'ordinamento topologico $v_1, \ldots, v_n$

Digrafi aciclici

Lemma. Se G un ordinamento topologico, allora G è un DAG.

D. Ogni DAG ha un ordinamento topologico?

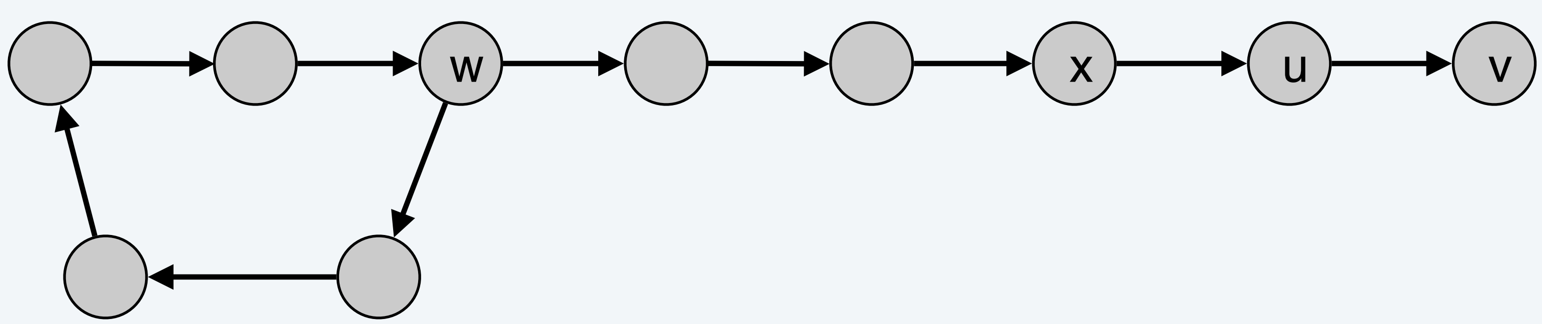
D. Se sì, come ne calcoliamo uno?

Digrafi aciclici

Lemma. Se G è un DAG, allora G ha un nodo senza archi entranti.

Dim. [per assurdo]

- Supponiamo che G sia un DAG e che ogni nodo abbia almeno un arco entrante.
- Prendiamo un nodo v, e iniziamo a seguire archi a ritroso da v. Poiché v ha almeno un arco entrante (u, v) possiamo risalire a ritroso a u.
- Poiché u ha almeno un arco entrante (x, u), possiamo risalire a x.
- Ripetiamo fino a visitare uno stesso nodo, sia esso w, due volte.
- Sia C la sequenza di nodi incontrati tra visite successive a w. C è un ciclo. ▪



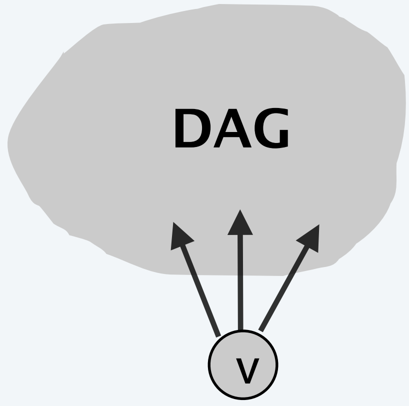
Digrafi aciclici

Lemma. Se G è un DAG, allora G ha un ordinamento topologico.

Dim. [per induzione su n]

- Caso base: vero se $n = 1$.
- Dato un DAG su $n > 1$ nodi, troviamo un nodo v senza archi entranti.
- $G - \{ v \}$ è un DAG, poiché la rimozione di v non può creare dei cicli.
- Per ipotesi induttiva, $G - \{ v \}$ ha un ordinamento topologico.
- Poniamo v all'inizio dell'ordinamento topologico; poi aggiungiamo nodi di $G - \{ v \}$ in ordine topologico. Ciò è corretto poiché v non ha archi entranti. ▪

To compute a topological ordering of G:
Find a node v with no incoming edges and order it first
Delete v from G
Recursively compute a topological ordering of $G-\{v\}$
  and append this order after v



Algoritmo di ordinamento topologico: tempo di esecuzione

Teorema. L'algoritmo trova un ordinamento topologico in tempo $O(m + n)$.

Dim.

- Manteniamo le seguenti informazioni:
  - $count(w)$ = numero rimanente di archi entranti in ciascun nodo w
  - S = insieme di nodi rimanenti senza archi entranti
- Inizializzazione: $O(m + n)$, con una singola scansione dell'intero grafo.
- Aggiornamento: per cancellare v
  - rimuovi v da S
  - decrementa $count(w)$ per tutti gli archi da v a w;
    e aggiungi w ad S se $count(w)$ diventa 0
  - il costo è $O(1)$ per arco. ▪