Ricerca Operativa
Modulo 2

Teoria dei Grafi: Parte 1
Introduzione

Marco A. Boschetti

Università degli Studi di Bologna
Dipartimento di Matematica
marco.boschetti@unibo.it

Outline

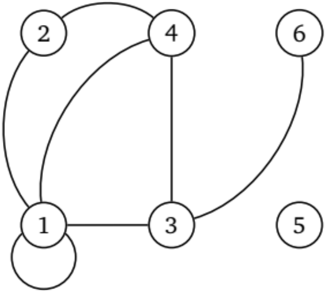
1. Introduzione alla Teoria dei Grafi
    - Definizioni di Base e Notazione
    - Applicazioni
    - Taglio di un grafo
    - Cammini, circuiti e cicli
    - Grafi parziali, sottografi e componenti
    - Alberi
    - Rappresentazione dei Grafi

Grafi non orientati e orientati

- Un grafo non orientato, rappresentato come $G = (V, E)$, è definito dall'insieme dei *vertici* (o *nodi*) e dall'insieme dei *lati* che congiungono coppie non ordinate di vertici:
  - $V = \{1, 2, \ldots, n\}$: insieme dei vertici (o nodi);
  - $E = \{e_1, e_2, \ldots, e_m\}$: insieme dei lati, che corrispondono a coppie *non ordinate* di vertici di V che sono *collegati*, i.e., un lato $e_k = \{i, j\}$ *collega* i vertici i e j.

- Un grafo orientato (o grafo diretto) $G = (V, A)$ si differenzia da un grafo non orientato per la sostituzione dell'insieme dei lati con l'insieme degli *archi*, che sono coppie *ordinate* di vertici:
  - $V = \{1, 2, \ldots, n\}$: insieme dei vertici (o nodi);
  - $A = \{a_1, a_2, \ldots, a_m\}$: insieme degli archi, che corrisponde a coppie *ordinate* di vertici di V, i.e., l'arco $a_k = (i, j)$ indica che il vertice i è collegato al vertice j.

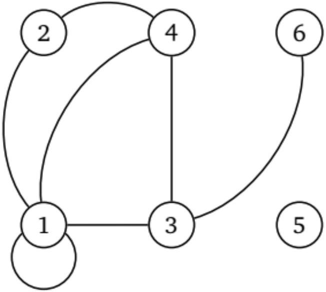
Esempio di grafo non orientato



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

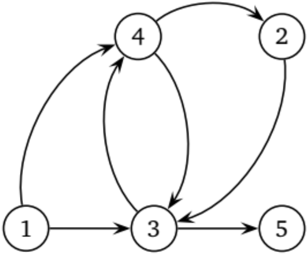
- Il lato $\{i, j\}$ *collega* i e j. Due vertici sono *adiacenti* se esiste il lato che li collega. Due lati sono *consecutivi* se hanno un vertice in comune.
- Il grafo ha un *loop* (lato $\{1, 1\}$), anche detto *autoanello* o *cappio*.

Esempio di grafo non orientato



- Si denota con $E(S)$ l'insieme dei lati con entrambi gli estremi nel sottoinsieme di vertici $S \subseteq V$ e $\Gamma(i)$ insieme dei vertici collegati a i.
- Se $S = \{1, 2, 4\}$ allora $E(S) = \{\{1, 1\}, \{1, 2\}, \{1, 4\}, \{2, 4\}\}$.
- $\Gamma(2) = \{1, 4\}$, $\Gamma(4) = \{1, 2, 3\}$, $\Gamma(5) = \emptyset$.

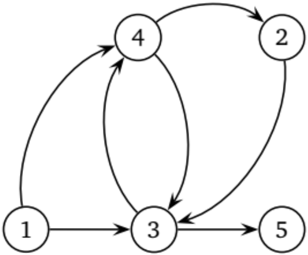
Esempio grafo orientato



$$V = \{1, 2, 3, 4, 5\} \quad A = \{(1,4), (1,3), (3,4), (4,3), (4,2), (2,3), (3,5)\}$$

- L'arco $(1, 4)$ *esce* dal vertice 1 e *entra* nel vertice 4.
- Dato l'arco (i, j) il vertice i è detto *vertice iniziale* (*coda* oppure *tail*) e j è detto *vertice terminale* (*testa* oppure *head*). Il vertice j è anche detto *successore* di i mentre i è detto *predecessore* di j.

Esempio grafo orientato

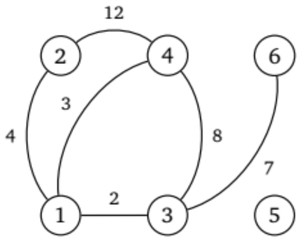


- Si denota con $A(S)$ l'insieme degli archi con entrambi gli estremi (vertice iniziale e finale) nel sottoinsieme di vertici $S \subseteq V$ e con $\Gamma^+(i)$ e $\Gamma^-(i)$ gli insiemi dei successori e dei predecessori di i.
- Se $S = \{1, 3, 4\}$, allora $A(S) = \{(1, 4), (1, 3), (3, 4), (4, 3)\}$.
- $\Gamma^+(1) = \{3, 4\}$, $\Gamma^+(4) = \{2, 3\}$, $\Gamma^+(5) = \emptyset$, mentre $\Gamma^-(1) = \emptyset$, $\Gamma^-(4) = \{1, 3\}$, $\Gamma^-(5) = \{3\}$.

Grafi pesati (non orientati e orientati)

- Il grafo G non orientato (orientato) è pesato sui lati (archi) se esiste una funzione $c : E \longrightarrow R$ ($c : A \longrightarrow R$) che associa un valore (o *peso*) ad ogni lato (arco).
- Il grafo G è pesato sui vertici se esiste una funzione $w : V \longrightarrow R$ che associa un valore (*peso*) ad ogni vertice.

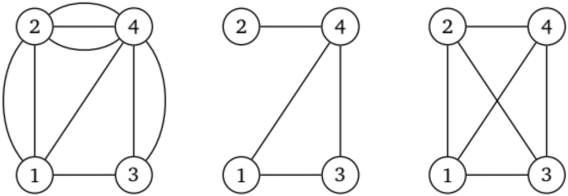
Esempio: grafo non orientato pesato



Grafi multipli, semplici e completi

- Un grafo è *multiplo* se può avere più di un lato per la stessa coppia di vertici.
- Un grafo è *semplice* se non comprende loop e lati multipli.
- Generalmente considereremo solo grafi semplici.
- Un grafo è *completo* se per ogni coppia di vertici esiste un lato.

Esempi



(a) Grafo multiplo (b) Grafo semplice (c) Grafo completo

Grafi: Applicazioni

Tra gli argomenti più noti nell'ambito della teoria dei grafi possiamo citare ad esempio:

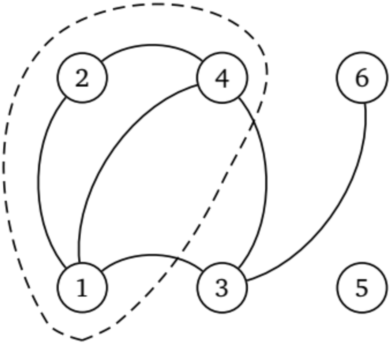
- Cammini Euleriani: originato dal problema posto da Eulero, per determinare un percorso che, partendo da una qualsiasi delle quattro zone della città di Könisberg, attraversasse tutti i sette ponti una ed una sola volta ritornando al punto di partenza.
- Colorazione dei grafi: dove un esempio di applicazione e la colorazione delle mappe per garantire di non usare lo stesso colore per nazioni confinanti.
- Problema della clique (cricca): per esempio per calcolare la clique (i.e., sottografo completo) di cardinalità massima.

Grafi: Applicazioni Reali (Reti Fisiche)

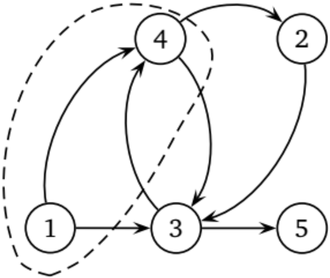
Applications	Physical Analog of Nodes	Physical Analog of Arcs	Flow
Communication systems	Telephone exchanges, computers, transmission facilities, satellites	Cables, fiber optic links, microwave relay links	Voice messages, data, video transmissions
Hydraulic systems	Pumping stations, reservoirs, lakes	Pipelines	Water, gas, oil, hydraulic fluids
Integrated computer circuits	Gates, registers, processors	Wires	Electrical current
Mechanical systems	Joints	Rods, beams, springs	Heat, energy
Transportation systems	Intersections, airports, rail yards	Highways, railbeds, airline routes	Passengers, freight, vehicles, operators

Taglio di un grafo

- Dato un sottoinsieme S di vertici, si dice *taglio* l'insieme dei lati (o archi) che congiungono i vertici in S con quelli in $V \setminus S$.



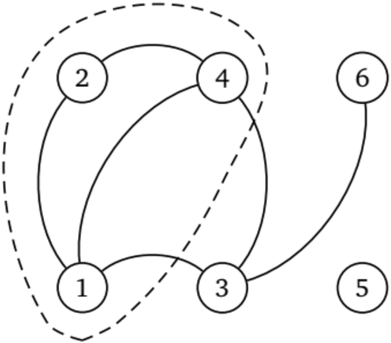
(a) Grafo non orientato



(b) Grafo orientato

Taglio di un grafo non orientato

- Per i grafi non orientati:
  $\delta_G(S) = \{\{i,j\} \in E : i \in S, j \in V \setminus S \text{ oppure } j \in S, i \in V \setminus S\}$.

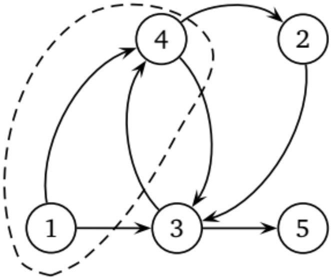


Nell'esempio $\delta_G(\{1,2,4\}) = \{(1,3),(3,4)\}$.

Taglio di un grafo orientato

- Nei grafi orientati distinguiamo tra archi uscenti ed entranti in $S \subset V$:
  - $\delta_G^+(S) = \{(i,j) \in A : i \in S, j \notin S\}$;
  - $\delta_G^-(S) = \{(i,j) \in A : j \in S, i \notin S\}$.

  Si noti che $\delta_G^+(S) \equiv \delta_G^-(V \setminus S)$.



Nell'esempio $\delta_G^+(\{1,4\}) = \{(1,3),(4,2),(4,3)\}$ e $\delta_G^-(\{1,4\}) = \{(3,4)\}$.

Cammini

- Un cammino è una sequenza di vertici $v_1, v_2, \ldots, v_k \in V$ tale che per ogni coppia di vertici consecutivi (v_i, v_{i+1}) esiste il corrispondente lato (grafo non orientato) o arco (grafo orientato).
- Un cammino P si può rappresentare sia come una sequenza di vertici:

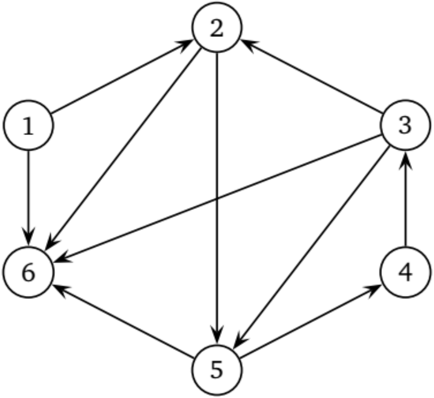
$$P = (v_1, v_2, v_3, \ldots, v_k)$$

- Un cammino può essere rappresentato anche come una sequenza archi (o lati):

$$P = ((v_1, v_2), (v_2, v_3), (v_3, v_4), \ldots, (v_{k-1}, v_k))$$

- In generale non ci sono vincoli che impediscono di visitare più volte alcuni vertici o percorrere più volte alcuni archi (o lati).

Cammini: Esempi



Esempi di cammini:

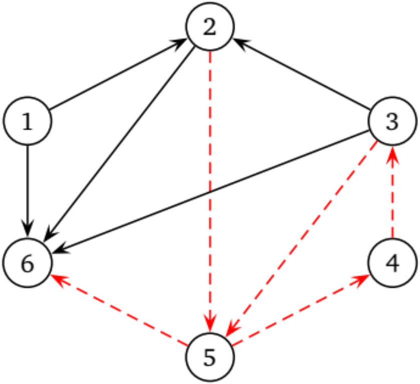
$P_1 = (2, 5, 4, 3, 5, 6)$

$P_2 = (1, 2, 5, 4, 3)$

$P_3 = (1, 2, 5, 4, 3, 2, 5)$

$P_4 = (3, 5, 4, 3)$

Cammini: Esempi



Esempi di cammini:

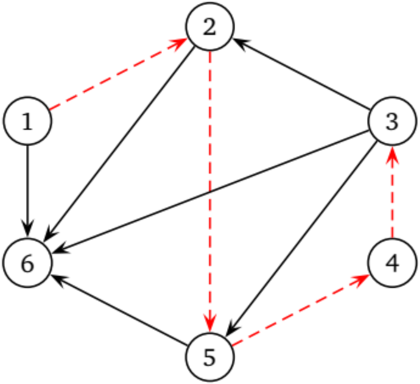
$P_1 = (2, 5, 4, 3, 5, 6)$

$P_2 = (1, 2, 5, 4, 3)$

$P_3 = (1, 2, 5, 4, 3, 2, 5)$

$P_4 = (3, 5, 4, 3)$

Cammini: Esempi



Esempi di cammini:

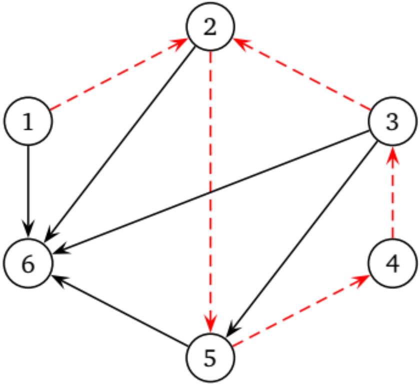
$P_1 = (2, 5, 4, 3, 5, 6)$

$P_2 = (1, 2, 5, 4, 3)$

$P_3 = (1, 2, 5, 4, 3, 2, 5)$

$P_4 = (3, 5, 4, 3)$

Cammini: Esempi



Esempi di cammini:

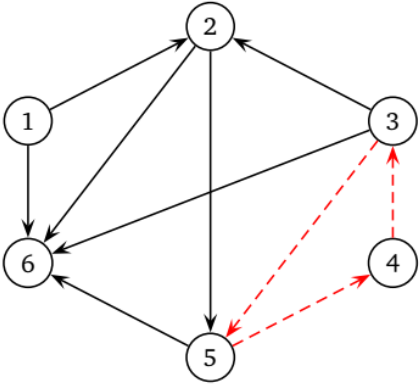
$P_1 = (2, 5, 4, 3, 5, 6)$

$P_2 = (1, 2, 5, 4, 3)$

$P_3 = (1, 2, 5, 4, 3, 2, 5)$

$P_4 = (3, 5, 4, 3)$

Cammini:Esempi



Esempi di cammini:

$P_1 = (2, 5, 4, 3, 5, 6)$

$P_2 = (1, 2, 5, 4, 3)$

$P_3 = (1, 2, 5, 4, 3, 2, 5)$

$P_4 = (3, 5, 4, 3)$

Costo di un cammino

- Dato un cammino $P = (v_1, v_2, v_3, \dots, v_k)$ il suo *costo* $c(P)$ è dato da:

$$c(P) = \sum_{i=1}^{k-1} c_{v_i v_{i+1}}$$

  dove c_{ij} è il costo dell'arco (i,j).

- Il costo di un cammino, a seconda del contesto e dell'applicazione, è anche detto *lunghezza*, *peso*, etc.

- Per esempio, se il costo di ciascun arco (i,j) corrisponde al tempo necessario per spostarsi dalla località i alla località j, allora il costo del cammino corrisponde al tempo necessario per visitare le località v_i, $i = 1, \dots, k$, nell'ordine indicato dal cammino.

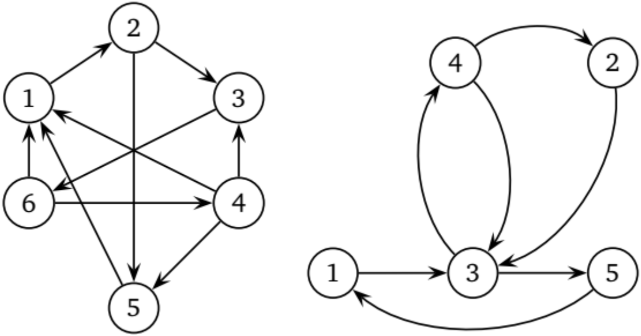
Cammini, circuiti e cicli

- Cammino semplice: non usa più di una volta lo stesso arco/lato. (P_1, P_2 e P_4 sono semplici; P_3 no)
- Cammino elementare: non passa più di una volta per lo stesso vertice. (P_2 è elementare; P_1, P_3 e P_4 no)
- Cammino hamiltoniano: usa una ed una sola volta tutti i vertici del grafo; quindi deve visitare tutti vertici del grafo.
- Cammino euleriano: usa una ed una sola volta tutti gli archi/lati del grafo.
- Circuito: in un grafo orientato è un cammino in cui il vertice iniziale coincide con il vertice terminale.
- Ciclo: controparte non orientata di un circuito.

Cammini, circuiti e cicli (2)

- Circuito elementare: è un circuito che, a parte il primo e l'ultimo vertice (che coincidono), non passa più di una volta per lo stesso vertice.
- Circuito hamiltoniano: è un circuito elementare che passa attraverso ogni vertice del grafo. Oppure, equivalentemente, è un cammino hamiltoniano chiuso (i.e., con un arco che collega l'ultimo vertice con il primo del cammino).
- Circuito euleriano: è un circuito elementare che passa attraverso ogni arco del grafo. Oppure, equivalentemente è un cammino euleriano chiuso.
- I grafi che possiedono almeno un circuito/ciclo hamiltoniano sono detti grafi hamiltoniani. Invece, i grafi che possiedono almeno un circuito/ciclo euleriano sono detti grafi euleriani.

Esempio di Circuiti

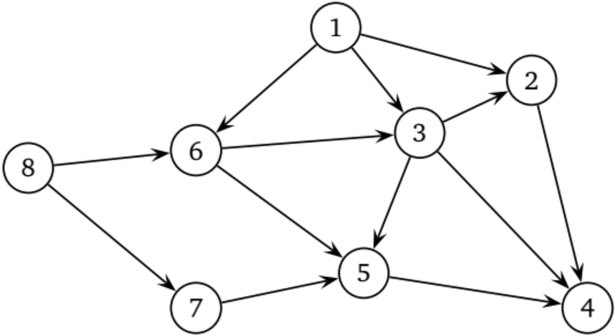


(a) Grafo hamiltoniano (b) Grafo non hamiltoniano

Circuito elementare (a) $C_1 = (1, 2, 3, 6, 1)$, (b) $C_2 = (3, 4, 2, 3)$.
Circuito hamiltoniano (a) $C_3 = (1, 2, 3, 6, 4, 5, 1)$, (b) non ne possiede.

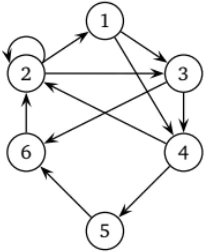
Grafi Aciclici

- Grafo aciclico: è un grafo che non contiene circuiti (cicli).

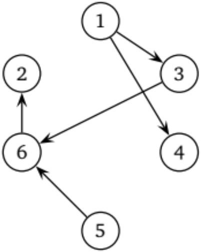


Grafi Parziali e Sottografi

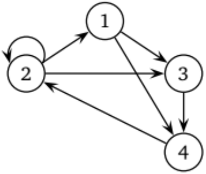
- Grafo parziale di $G = (V, A)$: è il grafo $G' = (V, A')$ dove $A' \subset A$.
- Sottografo di $G = (V, A)$: è il grafo $G' = (V', A')$ dove $V' \subseteq V$ e $A' \subseteq A$.



(a) Grafo orientato



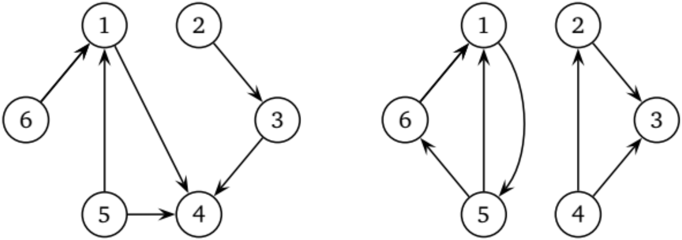
(b) Grafo parziale



(c) Sottografo

Connessioni e Componenti di un Grafo Orientato

- Grafo connesso: se il grafo *non orientato* relativo al grafo orientato ha almeno un cammino che congiunge ogni coppia di vertici.
- Se tale cammino non esiste allora il grafo viene detto disconnesso.



(a) Grafo connesso (b) Grafo disconnesso

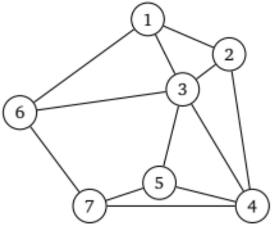
- Grafo fortemente connesso: se nel grafo esiste almeno un cammino orientato che congiunge ogni coppia di vertici.

Alberi

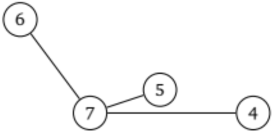
- Un grafo G_a non orientato di n vertici è un albero se rispetta le seguenti condizioni, che sono equivalenti:
  - G_a è connesso e aciclico;
  - G_a è aciclico e si crea un ciclo semplice se si aggiunge un lato al grafo G_a;
  - G_a è connesso, ma diventa non connesso non appena si elimina un solo lato di G_a;
  - G_a è connesso a ha $n-1$ lati;
  - G_a non ha cicli semplici e ha $n-1$ lati.
- In letteratura esistono anche altre condizioni equivalenti. Ognuna di queste definizioni equivalenti può essere utile a "identificare" e "utilizzare" gli alberi.

Alberi (2)

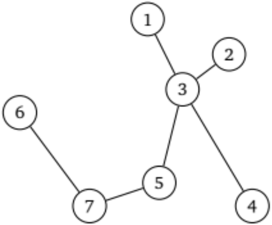
- Dato un grafo G, possono essere definiti dei sottografi di G che sono alberi. Tra questi, si definisce albero completo di G (detto anche spanning tree) un grafo parziale di G (i.e., "copre" tutti i vertici) che è un albero.
- Ogni grafo connesso ha almeno uno spanning tree.



(a) Grafo G



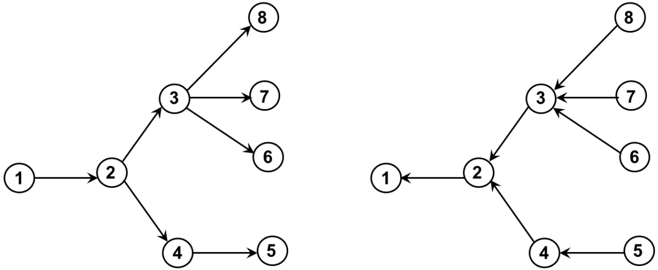
(b) Albero



(c) Albero completo

Alberi (3)

- Directed-out-tree: Albero in cui l'unico cammino dal nodo s a tutti gli altri nodi è diretto.
- Directed-in-tree: Albero in cui l'unico cammino da un qualsiasi altro nodo al nodo s è diretto.



(a) Directed-Out-Tree (b) Directed-In-Tree

Rappresentazione dei Grafi

- Il ruolo delle strutture dati è cruciale nello sviluppo di algoritmi efficienti.
- Il modo in cui sono salvati i dati del grafo (*rete*) nella memoria del calcolatore determina le performance degli algoritmi che operano su tali dati.
- Alcune delle operazioni che devono essere svolte dagli algoritmi sono le seguenti:
    - Accedere alle informazioni dei vertici;
    - Accedere alle informazioni degli archi;
    - Determinare tutti gli archi che partono da un vertice i;
    - Determinare tutti gli archi che arrivano a un vertice i;
    - Determinare tutti gli archi che incidono su un vertice i.

Rappresentazione dei Grafi (2)

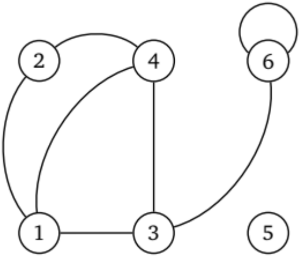
- In letteratura sono presentate numerose proposte. Alcune permettono un efficiente accesso ai dati, ma sono dispendiose dal punto di vista dell'occupazione di memoria, altre forniscono efficaci compromessi.
- La scelta della struttura dati più opportuna dipende principalmente dall'algoritmo che si deve implementare e dalle "risorse" a disposizione.

Rappresentazione dei Grafi: Matrice di Adiacenza

Definizione. La matrice di adiacenza Q di un grafo non orientato semplice $G = (V, E)$ è la matrice simmetrica $|V| \times |V|$ con elementi:

$$q_{ij} = \begin{cases} 1 & \text{se } \{i,j\} \in E; \\ 0 & \text{altrimenti.} \end{cases}$$

Esempio



(a) Grafo G

$$Q = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \to \Gamma(1) \\ \to \Gamma(2) \\ \to \Gamma(3) \\ \to \Gamma(4) \\ \to \Gamma(5) \\ \to \Gamma(6) \end{array}$$

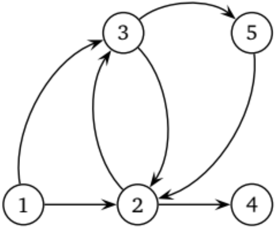
(b) Matrice di adiacenza

Rappresentazione dei Grafi: Matrice di Adiacenza

Definizione. La matrice di adiacenza Q di un grafo orientato semplice $G = (V, A)$ è la matrice $|V| \times |V|$ con elementi:

$$q_{ij} = \begin{cases} 1 & \text{se } (i,j) \in A; \\ 0 & \text{altrimenti.} \end{cases}$$

Esempio



(a) Grafo G

$$Q = \begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \left. \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{\substack{\downarrow \\ \Gamma^-(i)}} \right\} \to \Gamma^+(i) \end{array}$$

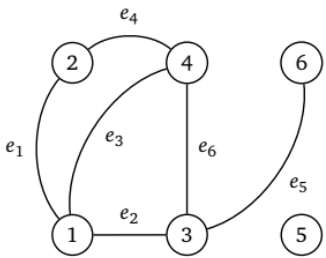
(b) Matrice di adiacenza

Rappresentazione dei Grafi: Matrice di Incidenza

Definizione. La matrice di incidenza nodi-lati D di un grafo non orientato $G = (V, E)$ è la matrice $|V| \times |E|$ con elementi:

$$d_{ik} = \begin{cases} 1 & \text{se il } k\text{-esimo lato è incidente nel vertice } i \text{ (i.e., } e_k = \{i, j\}); \\ 0 & \text{altrimenti.} \end{cases}$$

Esempio



(a) Grafo G

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

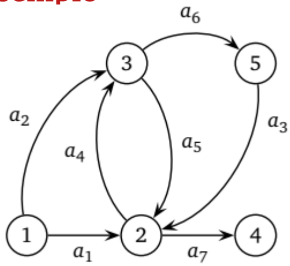
(b) Matrice di incidenza

Rappresentazione dei Grafi: Matrice di Incidenza

Definizione. La matrice di incidenza nodi-archi D di un grafo orientato $G = (V, A)$ è la matrice $|V| \times |A|$ con elementi:

$$d_{ik} = \begin{cases} 1 & \text{se il } k\text{-esimo arco esce dal vertice } i \text{ (i.e., } a_k = (i,j)); \\ -1 & \text{se il } k\text{-esimo arco entra nel vertice } i \text{ (i.e., } a_k = (j,i)); \\ 0 & \text{se } i \text{ non è vertice terminale di } a_k. \end{cases}$$

Esempio



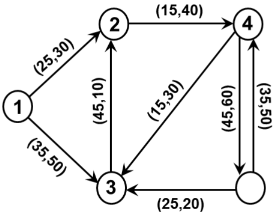
(a) Grafo G

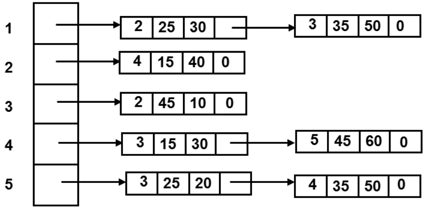
$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & -1 & 0 & 1 \\ 0 & -1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} \end{array}$$

(b) Matrice di incidenza

Rappresentazione dei Grafi: Liste di Adiacenza

- Le liste di adiacenza conservano per ogni vertice i la lista $A(i)$ degli archi che partono da esso.
- Per cui è necessario un vettore n-dimensionale ($n = |V|$) *first*, dove *first*(i) memorizza il puntatore al primo elemento della lista $A(i)$.



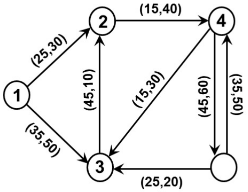


- Impiegando le liste di adiacenza si risparmia tempo calcolo e spazio di memoria. Però richiedono una “gestione” più complessa.

Rappresentazione dei Grafi: Forward Star

- La rappresentazione forward star richiede di salvare le informazioni degli archi in un vettore m-dimensionale ($m = |A|$). Gli archi devono essere ordinati per indice del *vertice iniziale* (*tail node*) crescente.

- Un vettore n-dimensionale ($n = |V|$) di puntatori *point* memorizza l'indice in cui sono salvate le informazioni del primo arco che *parte* dal vertice i nel corrispondente vettore. Gli archi che *partono* dal vertice i sono posizionati da $point(i)$ fino a $point(i+1)-1$.



	point
1	1
2	3
3	4
4	5
5	7
6	9

	tail	head	cost	capacity
1	1	2	25	30
2	1	3	35	50
3	2	4	14	40
4	3	2	45	10
5	4	3	15	30
6	4	5	45	60
7	5	3	25	20
8	5	4	35	50

Rappresentazione dei Grafi: Forward e Backward Star

- La rappresentazione forward star consente di accedere in modo efficiente agli archi che partono da un determinato vertice i.
- Nel caso sia necessario accedere agli archi che arrivano a un vertice i, la forward star non permette un'equivalente performance.
- Nel caso l'algoritmo necessiti di accedere agli archi che arrivano a un determinato vertice i è necessario utilizzare la backward star.
- La rappresentazione backward star è analoga alla forward star, ma gli archi sono ordinati per indice del *vertice finale* (*head node*) crescente;
- Inoltre, il vettore $point(i)$ memorizza l'indice in cui sono salvate le informazioni del primo arco che *arriva* al vertice i nel corrispondente vettore. Gli archi che arrivano al vertice i sono posizionati da $point(i)$ fino a $point(i+1)-1$.