Appendice B

Grafi e Reti

In questa appendice richiamiamo i principali concetti relativi a grafi e reti; descriviamo inoltre alcune classi di strutture dati che possono essere utilizzate per implementare efficientemente algoritmi su grafi e reti.

B.1 I grafi: notazione e nomenclatura

Un grafo, $G = (N, A)$, è una coppia di insiemi: N è un insieme finito e non vuoto di elementi, mentre A è un insieme finito di coppie di elementi distinti di N.

B.1.1 Grafi, nodi, archi

N è detto *insieme dei nodi* (o *vertici*), e usualmente viene indicato mediante i primi $n = |N|$ numeri naturali: $N = \{1, 2, \ldots, n\}$. L'insieme A, detto *insieme degli archi*, è in generale indicato con $A = \{a_1, a_2, \ldots, a_m\}$ ($m = |A|$). Nel seguito un arco verrà indifferentemente denotato o da un nome che lo individua, ad esempio a_k, oppure da una coppia di nodi, ad esempio (i, j); se la coppia non è ordinata, cioè (i, j) e (j, i) rappresentano lo stesso arco, allora l'arco è detto *non orientato*, altrimenti è detto *orientato* e in tal caso (i, j) e (j, i) rappresentano due archi diversi. I nodi i e j sono detti *estremi* dell'arco (i, j), che è *incidente* in essi; in questo caso si dirà che i nodi i e j sono *adiacenti*. Se l'arco è orientato, allora i è la *coda* e j è la *testa* di (i, j); l'arco è *uscente* da i e *entrante* in j. Un grafo i cui archi sono tutti non orientati è detto *non orientato* o *simmetrico*, ed è detto *orientato* se tutti i suoi archi sono orientati.

In un certo senso i grafi orientati sono una generalizzazione dei grafi non orientati: infatti un arco non orientato può essere rappresentato per mezzo di una coppia di archi orientati, come indicato in figura B.1. In figura B.2 sono rappresentati un grafo non orientato (a) ed un grafo orientato (b). Nel seguito pertanto i principali concetti verranno dati solo in termini di

grafi orientati in quanto la loro estensione ai grafi non orientati è in genere immediata.

Figura B.1: Equivalenza tra un arco non orientato ed una coppia di archi orientati

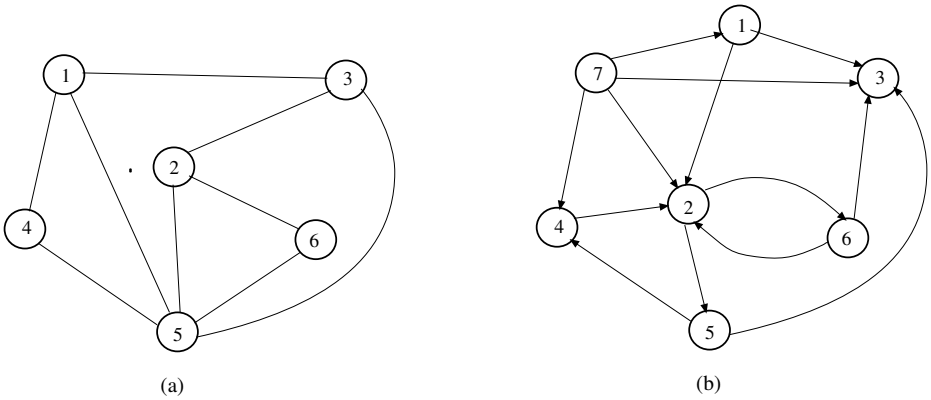


Figura B.2: Esempi di grafi

Esercizio B.1 *Indicare i nodi adiacenti al nodo* 5 *e gli archi incidenti nel nodo* 1 *del grafo in figura B.2(a).*

Esercizio B.2 *Indicare i nodi adiacenti al nodo* 5 *e gli archi incidenti nel nodo* 1 *del grafo in figura B.2(b).*

Una *rete* è un grafo ai cui nodi e/o archi sono associati dei pesi. Tali pesi possono avere significati diversi a seconda del contesto; ad esempio, se il grafo rappresenta una rete idraulica, i pesi associati agli archi possono rappresentare la portata (numero di litri per unità di tempo) e il flusso (effettiva quantità di acqua che fluisce nell'arco), mentre i pesi associati ai nodi possono rappresentare la quantità di acqua immessa nella rete o da essa estratta. I grafi rappresentano la struttura topologica delle reti; nel seguito faremo uso indifferentemente dei termini 'grafo' e 'rete'.

Dato un grafo (orientato o non orientato), per ogni nodo $i \in N$, si indica con $N(i)$ l'*insieme dei nodi adiacenti* ad i e con $S(i)$ l'*insieme degli archi incidenti* in i; $|S(i)|$ è detto il *grado* del nodo i ed è indicato con g_i.

Dato un grafo orientato, per ogni nodo $i \in N$, si indica con $FN(i)$ e $BN(i)$ rispettivamente l'*insieme dei nodi successori* e l'*insieme dei nodi predecessori*:

$$\begin{aligned} FN(i) &= \{j \in N : \exists (i,j) \in A\}, \\ BN(i) &= \{j \in N : \exists (j,i) \in A\}; \end{aligned}$$

mentre con $FS(i)$ e $BS(i)$ si indicano rispettivamente l'*insiemi degli archi uscenti* da i, o *stella uscente* di i, e l'*insieme degli archi entranti* in i, o *stella entrante* di i:

$$\begin{aligned} FS(i) &= \{(x,y) \in A : x = i\}, \\ BS(i) &= \{(x,y) \in A : y = i\}. \end{aligned}$$

Valgono le relazioni $FN(i) \cup BN(i) = N(i)$ e $FS(i) \cup BS(i) = S(i)$; $|FN(i)| = |FS(i)|$ e $|BN(i)| = |BS(i)|$ sono detti rispettivamente *grado uscente* e *grado entrante* di i.

Esercizio B.3 *Indicare la stella uscente e la stella entrante del nodo* 2 *del grafo in figura B.2(b).*

Dato un grafo $G = (N, A)$, il grafo $G' = (N, A')$, con $A' \subset A$, è detto *grafo parziale* di G; il grafo $G'' = (N'', A'')$, con $N'' \subset N$ e $A'' = \{(i,j) \in A : i, j \in N''\}$, è detto *grafo indotto* da N''. Un grafo $G^* = (N^*, A^*)$, è un *sottografo* di G se A^* è contenuto in A e N^* contiene tutti i nodi estremi degli archi di A^*.

Esercizio B.4 *Disegnare il grafo parziale* $G' = (N, A')$ *del grafo* $G = (N, A)$ *in figura B.2(b), dove* $A' = \{(1,2), (2,5), (2,6), (4,2), (5,3), (7,3)\}$.

Esercizio B.5 *Disegnare il grafo* $G'' = (N'', A'')$ *indotto da* $N'' = \{2, 4, 5, 6\}$ *sul grafo* $G = (N, A)$ *in figura B.2(b).*

Esercizio B.6 *Disegnare il sottografo* $G^* = (N^*, A^*)$ *del grafo* $G = (N, A)$ *in figura B.2(b), dove* $A^* = \{(1,3), (7,2), (7,3)\}$.

B.1.2 Cammini, cicli

Dato un grafo orientato G, un *cammino* tra il nodo i_0 ed il nodo i_q è una sequenza di nodi e di archi $C = \{i_0, a_1, i_1, a_2, i_2, \ldots, i_{q-1}, a_q, i_q\}$, in cui per ogni $h = 1, \ldots, q$, a_h è incidente in i_{h-1} e i_h. Se, per $h = 1, \ldots, q$, è $a_h = (i_{h-1}, i_h)$, allora C è un *cammino orientato* da i_0 a i_q.

Il nodo i_0 è detto l'*origine* e i_q è detto la *destinazione* di C. Una sottosequenza di C è detta *sottocammino*. Se $i_0 = i_q$, C è detto un *ciclo* (eventualmente orientato). Un cammino (ciclo) non contenente cicli come sottocammini (propri) è detto un *cammino* (*ciclo*) *semplice* (eventualmente orientato). Nei cammini (cicli) che non sono semplici vi è ripetizione di nodi; essi possono avere anche ripetizioni di archi (vedi Fig B.3). Quando non si ingenerano equivoci, un cammino può essere descritto mediante la sola sequenze dei suoi nodi o dei suoi archi.

Esercizio B.7 *Individuare un cammino tra i nodi* 2 *e* 4 *sul grafo* $G = (N, A)$ *in figura B.2(a); dire se è semplice.*

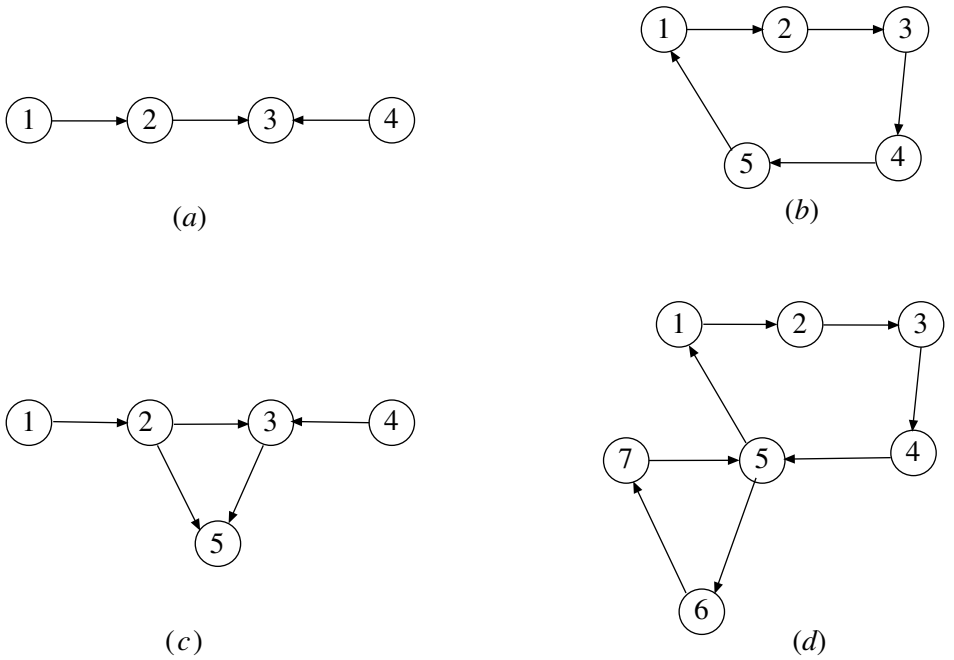


Figura B.3: (a) un cammino semplice; (b) un ciclo orientato semplice; (c) un cammino non semplice con ripetizione di archi; (d) un ciclo orientato non semplice senza ripetizione di archi

Esercizio B.8 *Individuare un ciclo orientato semplice formato da tre nodi e tre archi sul grafo* $G = (N, A)$ *in figura B.2(b).*

Un ciclo semplice formato da n archi è detto *ciclo Hamiltoniano*; esso passa per ogni nodo del grafo una e una sola volta. Un ciclo senza ripetizione di archi formato da m archi è detto *ciclo Euleriano*; esso passa attraverso ciascun arco del grafo una e una sola volta. Si può dimostrare che un grafo possiede un ciclo Euleriano se e solo se il grado di ogni nodo è pari, e che un grafo orientato possiede un ciclo Euleriano orientato se e solo se le cardinalità della stella uscente e della stella entrante di ogni nodo sono uguali.

Esercizio B.9 *Dimostrare l'affermazione precedente.*

Esercizio B.10 *Individuare, se esiste, un ciclo Hamiltoniano sul grafo* $G = (N, A)$ *in figura B.2(a).*

Due nodi i e j sono *connessi* se esiste un cammino tra di essi; in un grafo orientato si dice che j è *connesso a* i se esiste un cammino orientato da i a j.

Esercizio B.11 *Elencare tutti i nodi che sono connessi al nodo* 2 *sul grafo* $G = (N, A)$ *in figura B.2(b).*

B.1.3 Tagli e connettività

Dato un grafo $G = (N, A)$, una partizione di N in due sottoinsiemi non vuoti N' e N'' è detta un *taglio* del grafo e viene indicata con (N', N''); l'insieme degli archi aventi un estremo in N' e l'altro in N'' è detto *insieme degli archi del taglio* e verrà denotato con $A(N', N'')$:

$$A(N', N'') = \{(i, j) \in A : i \in N', j \in N'', \text{ oppure } j \in N', i \in N''\} .$$

Esercizio B.12 *Sia dato il taglio* $(\{1, 3, 5\}, \{2, 4, 6\})$ *del grafo* $G = (N, A)$ *in figura B.2(a), fornire l'insieme degli archi del taglio.*

Se il grafo è orientato, per ogni taglio si possono distinguere due insiemi di archi del taglio, l'insieme $A^+(N', N'')$ detto degli *archi diretti* del taglio e l'insieme $A^-(N', N'')$ detto degli *archi inversi* del taglio:

$$\begin{aligned} A^+(N', N'') &= \{(i, j) \in A : i \in N', j \in N''\}, \\ A^-(N', N'') &= \{(i, j) \in A : j \in N', i \in N''\}. \end{aligned}$$

Ovviamente, $A(N', N'') = A^+(N', N'') \cup A^-(N', N'')$.

Esercizio B.13 *Sia dato il taglio del grafo* $G = (N, A)$ *in figura B.2(b) individuato dagli insiemi* $N' = \{1, 3, 5, 7\}$ *e* $N'' = \{2, 4, 6\}$*, fornire gli insiemi* $A^+(N', N'')$ *e* $A^-(N', N'')$*.*

Utilizzando il concetto di taglio possiamo ridefinire la relazione di connessione fra nodi. Due nodi i e j sono connessi se non esiste un taglio (N', N'') tale che sia $i \in N'$, $j \in N''$ e $A(N', N'') = \emptyset$.

Esercizio B.14 *Dimostrare l'equivalenza delle due definizioni di connessione, basate rispettivamente sui cammini e sui tagli.*

Analogamente, in un grafo orientato j è connesso a i se non esiste un taglio (N', N'') tale che sia $i \in N'$, $j \in N''$ e $A^+(N', N'') = \emptyset$.

Un *grafo connesso* è un grafo in cui tutte le coppie di nodi sono connesse, altrimenti è detto *non connesso*. Un *grafo fortemente connesso* è un grafo in cui per ogni coppia di nodi i, j si ha che j è connesso a i, cioè esiste un cammino orientato da i a j. Ad esempio tutti i grafi in figura B.3 sono connessi, mentre solamente i grafi (b) e (d) sono fortemente connessi.

Dato un grafo non connesso $G = (N, A)$, ogni grafo connesso indotto da un sottoinsieme di N, massimale rispetto alla connessione, è detto *componente connessa* di G. Ad esempio il grafo di figura B.4 contiene due componenti connesse $G' = (N', A')$ e $G'' = (N'', A'')$, con $N' = \{1, 2, 3\}$, $A' = \{(1, 2), (1, 3), (2, 3)\}$, $N'' = \{4, 5, 6, 7\}$, $A'' = \{(4, 7), (6, 5), (6, 7), (7, 5)\}$.

Un grafo è detto *completo* se esiste un arco tra ogni coppia di nodi. Pertanto un grafo non orientato completo possiede $m = n(n-1)/2$ archi, mentre un grafo orientato completo possiede $m = n(n-1)$ archi.

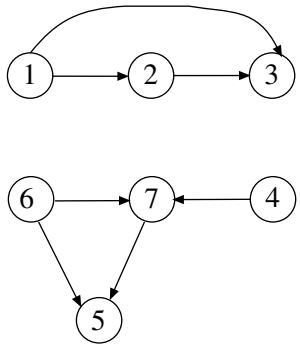


Figura B.4: Un grafo formato da due componenti connesse

Esercizio B.15 *Disegnare un grafo completo non orientato con 4 nodi e un grafo completo orientato con tre nodi.*

Un grafo $G = (N, A)$ è detto *bipartito* se esiste un taglio (N', N'') il cui insieme degli archi $A(N', N'')$ coincide con A, cioè se è possibile partizionare N in N' e N'' in modo tale che tutti gli archi abbiano un estremo in N' e l'altro in N''. Un grafo (orientato) bipartito è detto completo se esiste un arco (orientato) per ogni coppia di nodi non appartenenti al medesimo sottoinsieme del taglio; indicando con $n' = |N'|$ e $n'' = |N''|$ le cardinalità dei due sottoinsiemi, il numero di archi è $m = n'n''$ se il grafo è non orientato, mentre è $m = 2n'n''$ nel caso sia orientato.

Esercizio B.16 *Dimostrare le affermazioni fatte sul numero di archi dei grafi completi e dei grafi bipartiti completi (orientati e non).*

Esercizio B.17 *Disegnare un grafo bipartito completo non orientato con 8 nodi in cui $n' = 3$ e $n'' = 5$.*

B.1.4 Alberi

Un grafo connesso e privo di cicli è detto *albero*. Un albero $T = (N, A)$, con $|N| = n$, è tale che $|A| = n - 1$. Sono equivalenti alla precedente le seguenti definizioni: un albero è un grafo connesso con $n - 1$ archi; un albero è un grafo privo di cicli con $n - 1$ archi.

Esercizio B.18 *Dimostrare l'equivalenza delle definizioni date.*

Un *albero radicato* è un albero in cui sia stato selezionato un nodo, detto *radice dell'albero*; in un tale albero i nodi possono essere ordinati per "livelli" in modo ricorsivo: la radice è posta al livello 0 e i nodi adiacenti ad essa sono posti al livello 1; al livello $k + 1$ appartengono i nodi che non appartengono al livello $k - 1$ e che sono adiacenti ai nodi del livello k. Nel grafo di figura B.5, se si assume il nodo 1 come radice, al livello 0 appartiene il nodo 1, al

livello 1 si trovano i nodi 2 e 3, a livello 2 si trovano i nodi 4, 5, 6 e 7, mentre a livello 3 si trovano i nodi 8 e 9.

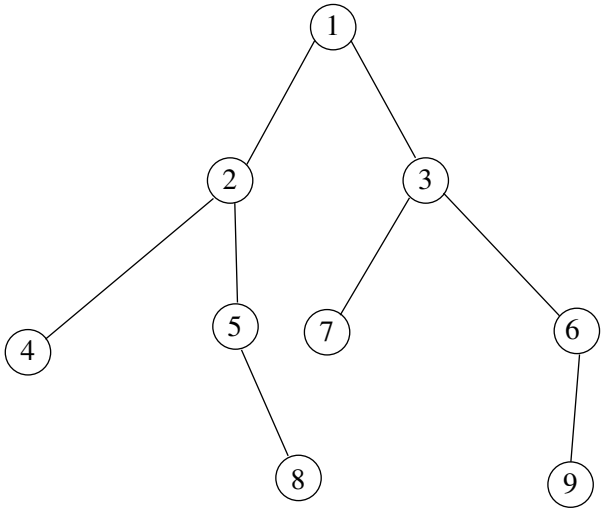


Figura B.5: Un albero

Esercizio B.19 *Disegnare l'albero in figura B.5 radicato nel nodo* 3*; individuare i nodi a livello* 2*.*

Ogni arco di un albero radicato connette due nodi appartenenti a livelli adiacenti. Per ciascun arco (i, j) con i a livello k e j a livello $k + 1$, i è detto *padre* di j e questi *figlio* di i; la radice non ha padre. Nodi aventi lo stesso padre sono detti *fratelli*. Un nodo senza figli è detto *foglia* dell'albero radicato.

Esiste un solo cammino tra la radice e qualsiasi nodo dell'albero; la lunghezza (in numero di archi) di tale cammino è uguale al livello cui appartiene il nodo destinazione del cammino. Un nodo i che appartiene al cammino dalla radice ad un nodo j è detto un *antenato* di j e questi un *discendente* di i (ogni nodo è antenato e discendente di sé stesso). Un sottoalbero $T(j)$ di un albero radicato è il grafo indotto dall'insieme dei nodi discendenti di j; in altri termini $T(j)$ è formato da j, da tutti gli altri suoi discendenti e da tutti gli archi dell'albero tra questi nodi.

Ad esempio, nell'albero di figura B.5, considerato radicato in 1, il padre di 3 è 1; i nodi 4 e 5 sono fratelli; l'insieme delle foglie è $\{4, 7, 8, 9\}$; l'insieme degli antenati di 3 è $\{1, 3\}$ e quello dei discendenti è $\{3, 6, 7, 9\}$. Inoltre, il sottoalbero $T(2)$, disegnato in figura B.6, dell'albero di figura B.5, radicato in 1, contiene i nodi $2, 4, 5$ e 8.

Esercizio B.20 *Individuare sull'albero disegnato per l'esercizio precedente i figli della radice, gli antenati di* 6 *e i discendenti di* 1*; disegnare il sottoalbero* $T(6)$*.*

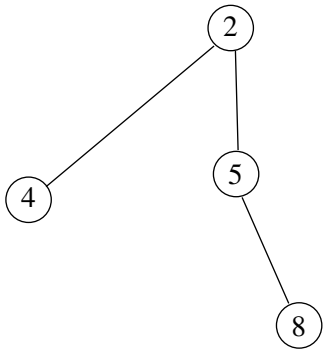


Figura B.6:

Un albero radicato, i cui archi sono orientati, è detto *orientato* se tutti i suoi archi sono orientati dal padre verso il figlio (o dal figlio verso il padre).

Dato un grafo $G = (N, A)$, un suo grafo parziale $T = (N, A_T)$ che sia un albero è detto *albero di copertura* (*spanning tree*) di G; nel grafo di figura B.7 è evidenziato un albero di copertura.

Ogni arco $(i, j) \in A$, non appartenente a T, forma con gli archi di T un unico ciclo che indicheremo con $C_T(i, j)$. Inoltre, l'eliminazione di un arco $(i, j) \in A_T$ divide l'albero T in due sottoalberi $T_i = (N_i, A_i)$ e $T_j = (N_j, A_j)$, individuando un taglio (N_i, N_j). Gli archi del taglio sono quelli dell'insieme

$$A(N_i, N_j) = \{(k, l) \in A : k \in N_i, l \in N_j \text{ oppure } l \in N_i, k \in N_j\} ;$$

cioè, essi sono (i, j) stesso e tutti gli archi non appartenenti a T che, quando aggiunti all'albero, formano un ciclo contenente (i, j).

Un grafo le cui componenti connesse sono alberi è detto *foresta*.

Esercizio B.21 *Disegnare una foresta del grafo in figura B.4.*

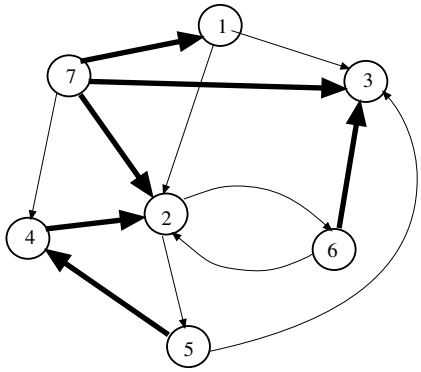


Figura B.7: Un grafo orientato e un suo albero di copertura

B.2 Rappresentazione di grafi ed alberi

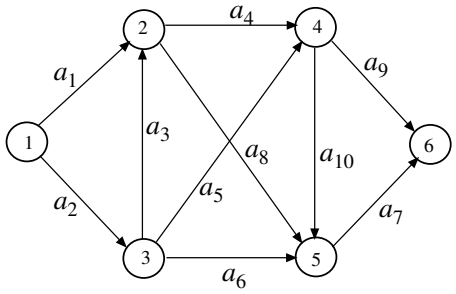
In questo paragrafo verranno presentate le strutture dei dati fondamentali per la rappresentazione dei grafi e delle reti, e verranno introdotti alcuni algoritmi elementari che serviranno come strumenti di base per la costruzione di algoritmi per problemi di ottimizzazione su reti.

B.2.1 Matrici di incidenza e liste di adiacenza

Dato un grafo orientato $G = (N, A)$, la sua *matrice di incidenza* $E = [e_{ik}]$ è una matrice $n \times m$ (le righe corrispondono ai nodi e le colonne agli archi), così definita:

$$e_{ik} = \begin{cases} -1, & \text{se } i \text{ è la coda dell'arco } \quad a_k, \\ 1, & \text{se } i \text{ è la testa dell'arco } \quad a_k, \\ 0, & \text{altrimenti.} \end{cases}$$

La matrice di incidenza ha due soli elementi diversi da 0 per ogni colonna: un -1 ed un 1. Un esempio di matrice di incidenza è riportato in figura B.8.



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	-1	-1	0	0	0	0	0	0	0	0
2	1	0	1	-1	0	0	0	-1	0	0
3	0	1	-1	0	-1	-1	0	0	0	0
4	0	0	0	1	1	0	0	0	-1	-1
5	0	0	0	0	0	1	-1	1	0	1
6	0	0	0	0	0	0	1	0	1	0

Figura B.8: Un grafo e la sua matrice di incidenza

La *lista di adiacenza* per stelle uscenti di un grafo orientato è la sequenza $\{FS(1), FS(2), \ldots, FS(n)\}$ degli insiemi degli archi uscenti dai nodi del grafo. Nell'esempio in figura B.8, la lista di adiacenza è $\{\{a_1, a_2\}, \{a_4, a_8\}, \{a_3, a_5, a_6\}, \{a_9, a_{10}\}, \{a_7\}, \emptyset\}$. In modo analogo si definisce la lista di adiacenza per stelle entranti $\{BS(1), BS(2), \ldots, BS(n)\}$.

Le liste di adiacenza consentono una efficiente memorizzazione dei grafi. Ad esempio in figura B.9 è riportata una possibile rappresentazione del grafo di figura B.8 che utilizza un sistema di liste. In particolare:

1. I nodi sono rappresentati per mezzo di una lista in cui ogni record corrisponde ad un nodo e contiene quattro campi: il nome del nodo, il puntatore al nodo successivo nella lista, il puntatore al primo arco della stella uscente, il puntatore al primo arco della stella entrante; un puntatore al primo nodo consente di iniziare la scansione.

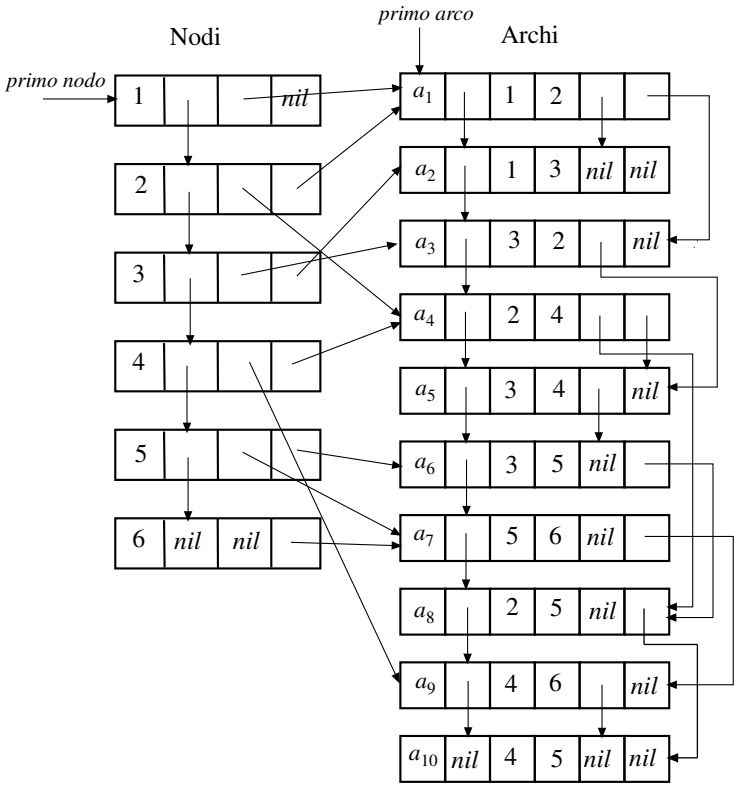


Figura B.9: Rappresentazione di un grafo per mezzo di liste di adiacenza

2. Gli archi sono rappresentati per mezzo di una lista in cui ogni record corrisponde ad un arco e contiene sei campi: il nome dell'arco, il puntatore all'arco successivo, la coda, la testa, il puntatore all'arco successivo della stella uscente cui esso appartiene, il puntatore all'arco successivo della stella entrante cui esso appartiene; anche qui c'è un puntatore al primo arco.

Tale struttura dati permette agevolmente sia la scansione dei nodi e degli archi, sia la scansione degli archi di una stessa stella uscente od entrante quando si conosca il nodo relativo. Essa permette anche inserimenti e rimozioni di nodi e/o di archi. La struttura dati può essere ulteriormente ampliata: ad esempio, a ciascun record di nodo e/o di arco possono essere aggiunti nuovi campi contenenti i pesi o le variabili di interesse. Un altro esempio di ampliamento della struttura consiste nell'introduzione di puntatori inversi dei puntatori sopra definiti; la presenza di tali puntatori consente la rimozione di un nodo o di un arco senza dover scorrere la lista alla ricerca del record precedente.

Esercizio B.22 *Scrivere una procedura che, utilizzando la struttura per li-*

ste di adiacenza descritta sopra, calcoli il grado entrante ed il grado uscente di ciascun nodo; se ne valuti la complessità.

Esercizio B.23 *Scrivere una procedura che, utilizzando la struttura per liste di adiacenza descritta sopra, inserisca un nuovo nodo $n+1$; se ne valuti la complessità.*

Esercizio B.24 *Scrivere una procedura che, utilizzando la struttura per liste di adiacenza descritta sopra, inserisca un nuovo arco a_{m+1}; se ne valuti la complessità.*

Esercizio B.25 *Scrivere una procedura che, utilizzando la struttura per liste di adiacenza descritta sopra, rimuova un nodo i e tutti gli archi incidenti in esso; se ne valuti la complessità.*

Esercizio B.26 *Risolvere il problema dell'esercizio precedente utilizzando una struttura ampliata con i puntatori inversi.*

Spesso, nella risoluzione di problemi su grafi, le rimozioni di nodi ed archi sono temporanee in quanto un particolare sottoproblema da risolvere è relativo ad una data porzione del grafo originario e sottoproblemi successivi sono relativi ad altre porzioni; non si vuole pertanto distruggere la struttura di dati che descrive il grafo originario. In tal caso, la rimozione fisica dei record relativi a nodi ed archi risulta inutilmente costosa. È più conveniente invece affiancare, in ciascun record, ai puntatori 'statici' che definiscono il grafo originario, nuovi puntatori 'dinamici' che definiscono il sottografo corrente, ed operare gli opportuni aggiornamenti su di essi.

Liste di adiacenza mediante vettori di puntatori

La struttura per liste di adiacenza può essere semplificata quando non si prevedono aggiunte o rimozioni di nodi e/o archi. In tal caso le liste a puntatori dei nodi e degli archi possono essere agevolmente realizzate mediante vettori facendo corrispondere l'indice del nodo o dell'arco con l'indice della componente del vettore contenente le informazioni relative al nodo o all'arco. Per realizzare la lista di adiacenza per stelle uscenti è sufficiente disporre di un vettore, $P\_FS[.]$, ad $n+1$ componenti, una per ogni nodo più una ausiliaria, e di un vettore, $H\_Arc[.]$, ad m componenti, una per ogni arco. L'elemento i-esimo del primo vettore ($i = 1, \ldots, n$) contiene il puntatore al primo arco della stella uscente del nodo i, mentre l'elemento $n+1$ punta all'arco fittizio $m+1$. L'elemento k-esimo del secondo vettore ($k = 1, \ldots, m$) contiene il nodo testa dell'arco k; gli archi sono ordinati per stelle uscenti.

Per conoscere la stella uscente del nodo i basta effettuare una scansione del vettore $H\_Arc[.]$ tra la posizione $P\_FS[i]$ e la posizione $P\_FS[i+1]-1$,

ottenendo le teste degli archi aventi i come coda. La stella uscente è vuota se $P\_FS[i] = P\_FS[i+1]$.

L'occupazione di memoria di questa rappresentazione della lista di adiacenza è $m + n + 1$.

Ad esempio per rappresentare il grafo di figura B.8 i due vettori $P\_FS[.]$ e $H\_Arc[.]$ assumono i seguenti valori:

i	1	2	3	4	5	6	7
$P\_FS[i]$	1	3	5	8	10	11	11

k	1	2	3	4	5	6	7	8	9	10
$H\_Arc[k]$	2	3	4	5	2	4	5	6	5	6

Analogamente si può realizzare la lista di adiacenza per stelle entranti.

Esercizio B.27 *Costruire la lista di adiacenza (per stelle entranti) del grafo in figura B.8.*

B.2.2 Rappresentazione di alberi: la funzione predecessore

Un albero radicato di radice r è rappresentato mediante la funzione predecessore p:

$$p_j = \begin{cases} i, & \text{se } i \text{ è padre di } j, \\ nil, & \text{se } j \text{ è la radice } (j = r). \end{cases}$$

Se gli archi dell'albero radicato sono archi orientati, e ci interessa memorizzare l'orientamento dell'arco che connette un nodo con il padre, basta porre $p_j = -i$, se i è padre di j e l'arco tra essi è (j, i).

Osserviamo che la funzione predecessore può essere convenientemente inserita nella struttura dati descritta nel paragrafo B.2.1, inserendo ulteriori campi in ciascun record della lista corrispondente all'insieme dei nodi. Infatti, per il record corrispondente al nodo i, è sufficiente aggiungere un campo contenente il puntatore al nodo p_i, un campo (booleano) per l'orientamento dell'arco (i, p_i) o (p_i, i) ed eventualmente un campo contenente il puntatore a tale arco nell'insieme degli archi.

B.2.3 Visite di un albero

Un'operazione particolarmente rilevante è la visita di un albero. A seconda dell'ordine con cui i nodi (e gli archi) vengono visitati avremo diversi tipi di visita.

Si dice *visita anticipata* di un albero $T = (N_T, A_T)$, di radice r e definito dalla funzione predecessore p, una visita dei nodi secondo la seguente regola: "un nodo i viene visitato solo se tutti i nodi appartenenti all'unico cammino in T tra r e i sono stati visitati", cioè un nodo può essere visitato solo dopo che sono stati visitati tutti gli altri suoi antenati. Pertanto la visita inzia dalla radice dell'albero e termina in una sua foglia. Osserviamo che la visita anticipata visita anche gli archi di T. Infatti, quando viene visitato un nodo

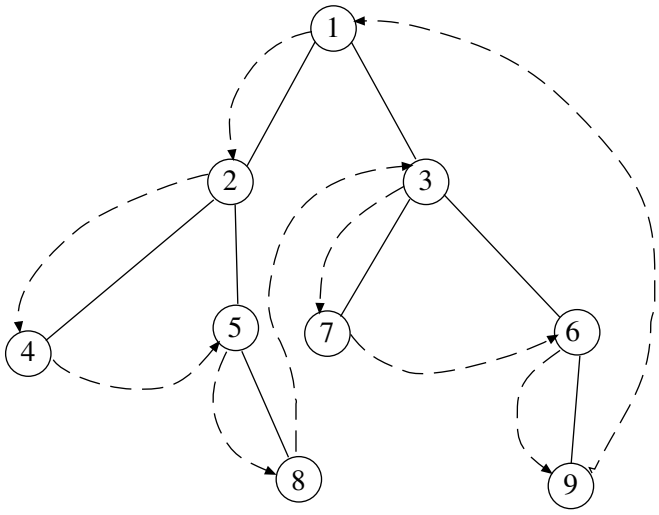


Figura B.10: La funzione $va(.)$ della visita anticipata

$i \neq r$, viene anche implicitamente visitato l'arco $(i, -p_i)$ (o (p_i, i)); quindi la visita anticipata induce un ordinamento sui nodi e sugli archi di T. Una visita anticipata è definita per mezzo di una funzione, $va(.)$, che associa ad ogni nodo i il nodo che verrà visitato dopo i attraverso una visita anticipata di T; inoltre, $va(.)$ associa all'ultimo nodo visitato il primo della visita. Dato un albero ci sono diverse funzioni $va(.)$ che realizzano una visita anticipata. In figura B.10 viene fornito un esempio di visita anticipata; essa permette di visitare consecutivamente i nodi di ciascun sottoalbero.

Definiamo *visita posticipata* di T una visita dei nodi secondo la seguente regola: "un nodo i viene visitato solo se tutti i suoi nodi figli sono stati visitati", cioè un nodo può essere visitato solo dopo che sono stati visitati tutti gli altri suoi discendenti. Analogamente alla funzione $va(.)$, possiamo definire la funzione di visita posticipata, $vp(.)$. Una particolare visita posticipata è data dalla funzione inversa di $va(.)$: $va^{-1}(j) = i \Leftrightarrow va(i) = j$. Con riferimento all'esempio in figura B.10 si ha:

i	1	2	3	4	5	6	7	8	9
$va(i)$	2	4	7	5	8	9	6	3	1
$va^{-1}(i)$	9	1	8	2	4	7	3	5	6

B.2.4 Livello dei nodi di un albero

La funzione *livello*, $lev(.)$, dei nodi di un albero associa ad ogni nodo i il suo livello, cioè il numero di archi dell'unico cammino nell'albero tra la radice r e i; $lev(.)$ può essere facilmente calcolata, date le funzioni predecessore p e visita anticipata $va(.)$, mediante la seguente procedura Livello:

```
Procedure Livello (r, p, va, lev):
    begin
        lev[r] := 0; u = va[r];
        while u ≠ r do
            begin lev[u] := lev[p[u]] + 1; u := va[u] end
    end.
```

Procedura 2.1: Calcolo della funzione *livello*

Esercizio B.28 *Determinare lev(.) per l'albero in figura B.10.*