GRAFI: RAPPRESENTAZIONE

[Deme, seconda edizione] cap. 12 sez. 12.2

Parametri di valutazione

Per valutare un approccio di rappresentazione, dobbiamo considerare due fattori importanti:

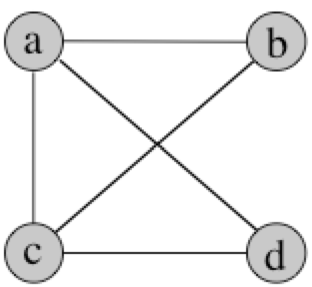
1. Lo **spazio** occupato dalla struttura dati
2. Il **costo computazionale** delle operazioni da effettuare su di essa. Nel caso dei grafi, le operazioni «semplici» sono:

`grado(vertice v)`	Ritorna il numero di archi incidenti al vertice v
`archiIncidenti(vertice v)`	Ritorna gli archi incidenti al vertice v
`sonoAdiacenti(vertici v e w)`	True se $(v, w) \in E$, false altrimenti
`aggiungiVertice(vertice v)`	Inserisce un nuovo vertice v nel grafo
`aggiungiArco(vertici v e w)`	Inserisce un nuovo arco tra i vertici v e w
`rimuoviVertice(vertice v)`	Rimuove il vertice v e tutti gli archi ad esso incidenti
`rimuoviArco(arco e)`	Rimuove l'arco e

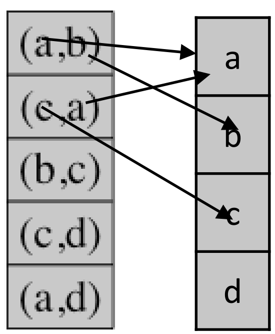
Lista di archi - rappresentazione

Rappresenta un grafo nella maniera più naturale, ispirandosi alla definizione $G = (V, E)$. Un insieme/lista di vertici V e un insieme/lista di archi E (con archi che puntano a vertici).

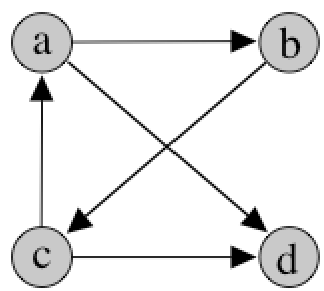
Dati n (numero di vertici) e m (numero di archi) lo spazio occupato è banalmente $\boldsymbol{O(m+n)}$.



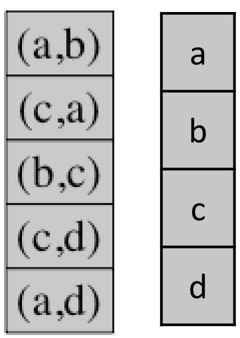
Grafo non orientato



Rappresentazione



Grafo orientato



Rappresentazione

Lista di archi – costo operazioni (grafi non orientati)

PRO: aggiunta vertici e archi.

CONTRO: tutte le altre operazioni dipendono da m (il numero degli archi).

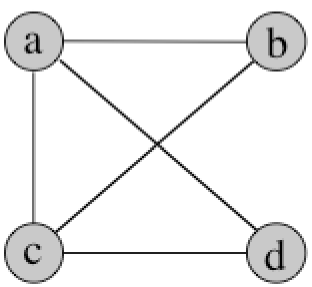
Operazione	**Tempo di esecuzione**
`grado`(v)	$O(m)$
`archiIncidenti`(v)	$O(m)$
`sonoAdiacenti`(x, y)	$O(m)$
`aggiungiVertice`(v)	$O(1)$
`aggiungiArco`(x, y)	$O(1)$
`rimuoviVertice`(v)	$O(m)$
`rimuoviArco`(e)	$O(m)$

Liste di adiacenza - rappresentazione

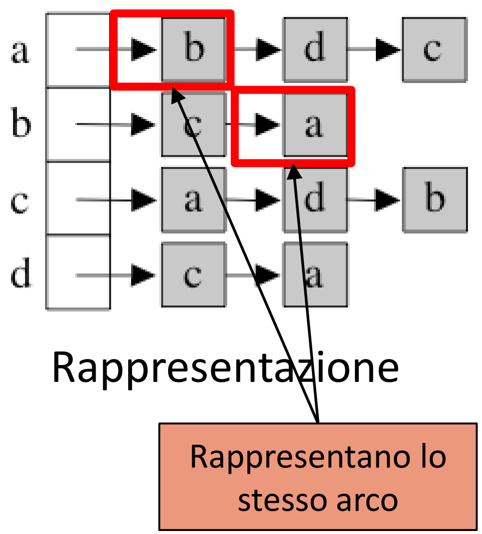
Ogni vertice v ha una lista contenente i vertici ad esso adiacenti.

Ogni vertice è rappresentato una sola volta, gli archi non sono esplicitamente rappresentati, ma abbiamo un elemento nelle liste (2 per grafi non orientati) per ogni arco. Occupa spazio $O(m+n)$.

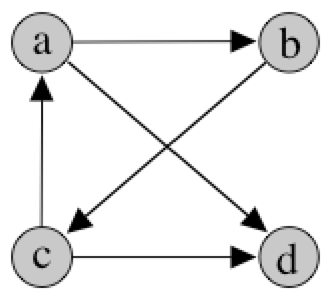
Adatta per grafi **sparsi** ($|E| \ll |V|^2$).



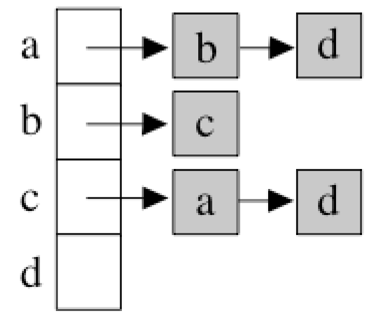
Grafo non orientato



Rappresentazione

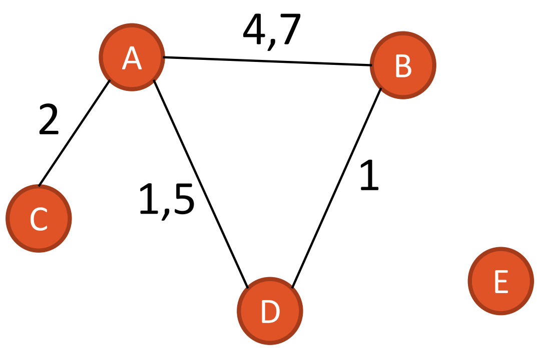


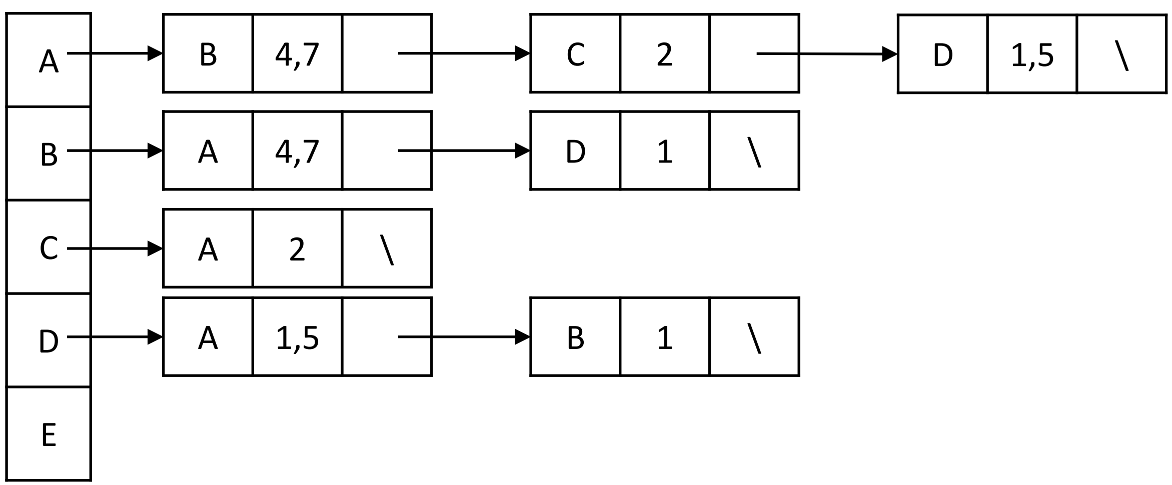
Grafo orientato



Rappresentazione

Liste di adiacenza – grafi pesati





Liste di adiacenza – costi operazioni

PRO: migliora l'accesso agli archi tramite vertici.

CONTRO: (per grafi non orientati) difficile mantenere la consistenza tra liste di adiacenza (es. rimuovi arco).

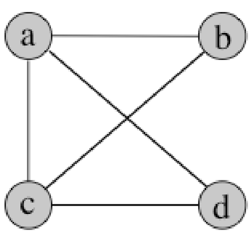
Operazione	Tempo di esecuzione
grado(v)	$O(\delta(v))$
archiIncidenti(v)	$O(\delta(v))$
sonoAdiacenti(x, y)	$O(\min\{\delta(x), \delta(y)\})$
aggiungiVertice(v)	$O(1)$
aggiungiArco(x, y)	$O(1)$
rimuoviVertice(v)	$O(m)$
rimuoviArco$(e = (x, y))$	$\boxed{O(\delta(x) + \delta(y))}$

Ricorda: δ(v) è il grado (numero di archi incidenti) di v

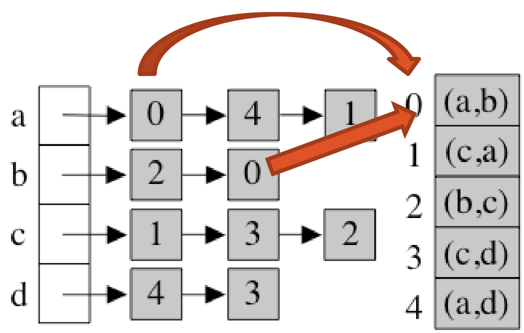
Liste di incidenza - rappresentazione

Ogni vertice v ha una lista contenente **un riferimento agli archi ad esso incidenti**. Gli archi (grafi non orientati) non sono duplicati, solo i riferimenti.

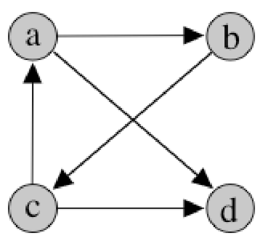
Occupa spazio $\boldsymbol{O(m+n)}$.



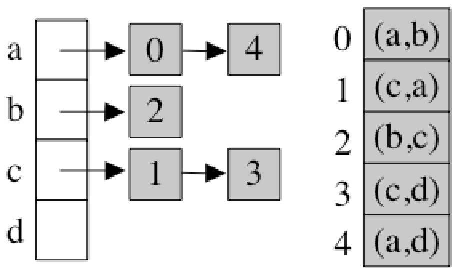
Grafo non orientato



Rappresentazione



Grafo orientato



Rappresentazione

Liste di incidenza – costi operazioni

Stesse considerazioni fatte per liste di adiacenza. Leggermente minore costo di «scansione» delle liste (liste di puntatori vs liste di vertici).

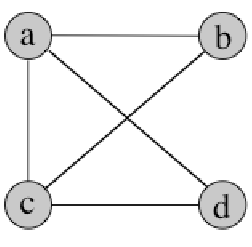
Operazione	**Tempo di esecuzione**
grado(v)	$O(\delta(v))$
archiIncidenti(v)	$O(\delta(v))$
sonoAdiacenti(x, y)	$O(\min\{\delta(x), \delta(y)\})$
aggiungiVertice(v)	$O(1)$
aggiungiArco(x, y)	$O(1)$
rimuoviVertice(v)	$O(m)$
rimuoviArco$(e = (x, y))$	$O(\delta(x) + \delta(y))$

Matrici di adiacenza - rappresentazione

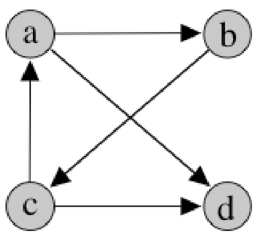
Il grafo è rappresentato tramite una matrice di interi di grandezza $n \times n$ (spazio occupato $\boldsymbol{O(n^2)}$).

Adatta per grafi **densi** ($|E| \approx |V|^2$).

$$M[v,w] = \begin{cases} 1 \ se \ (v,w) \in E \\ 0 \ altrimenti \end{cases}$$



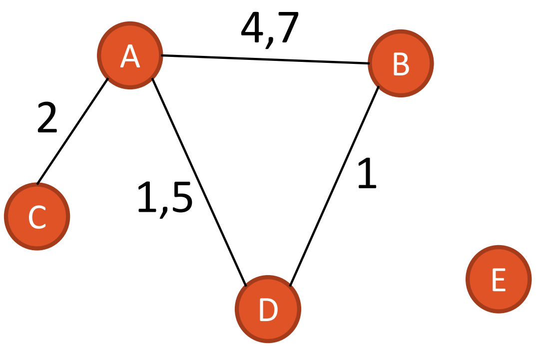
	a	b	c	d
a	0	1	1	1
b	1	0	1	0
c	1	1	0	1
d	1	0	1	0



	a	b	c	d
a	0	1	0	1
b	0	0	1	0
c	1	0	0	1
d	0	0	0	0

Nota: per i grafi non orientati
$M[v,w] = M[w,v]$
(la matrice è **simmetrica**).

Matrici di adiacenza - grafi pesati



	A	**B**	**C**	**D**	**E**
A	0	4,7	2	1,5	∞
B	4,7	0	∞	1	∞
C	2	∞	0	∞	∞
D	1,5	1	∞	0	∞
E	∞	∞	∞	∞	0

Matrici di adiacenza – costi operazioni

PRO: verifica di **presenza di archi** fatta in tempi rapidi.

CONTRO: le operazioni di aggiunta/rimozione di un vertice richiedono di operare su **tutta la matrice**.

Operazione	**Tempo di esecuzione**
`grado`(v)	$O(n)$
`archiIncidenti`(v)	$O(n)$
`sonoAdiacenti`(x, y)	$\boxed{O(1)}$
`aggiungiVertice`(v)	$\boxed{O(n^2)}$
`aggiungiArco`(x, y)	$\boxed{O(1)}$
`rimuoviVertice`(v)	$\boxed{O(n^2)}$
`rimuoviArco`(e)	$\boxed{O(1)}$

Domanda: se i nomi dei vertici non sono rappresentati con numeri naturali, è possibile mantenere complessità $O(1)$? Se sì, come?

Matrici di adiacenza – cammini

Una matrice di adiacenza rappresenta anche la **presenza di un cammino di lunghezza 1** tra ogni coppia di vertici v e w. In particolare, $v \underset{1}{\rightarrow} w$ se e solo se $M[v,w] \neq 0$.

È possibile sapere se esiste un cammino di lunghezza 2 tra v e w?

Moltiplichiamo M per se stessa

$$M^2[v,w] = \sum_{z \in V} (M[v,z] * M[z,w])$$

$M^2[v,w] \neq 0$ se e solo se esiste almeno un vertice z tale che $(v,z),(z,w) \in E$.

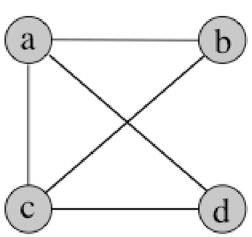
Che vuol dire che esiste un cammino di lunghezza 2 (che passa da z) tra v e w.

Si può dimostrare per induzione che **$M^n[v,w] \neq 0$ se e solo se esiste un cammino di lunghezza esattamente n tra v e w per ogni $n > 0$**.

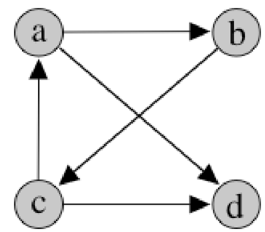
Matrici di incidenza

Il grafo è rappresentato tramite una matrice di interi di grandezza $n \times m$ (spazio occupato $\boldsymbol{O(n \times m)}$) in cui le righe indicizzano i vertici mentre le colonne indicizzano gli archi.

$$\boldsymbol{M[v,(x,y)\,]} = \begin{cases} \boldsymbol{1\ se\ v = x \wedge \langle v,y\rangle \in E} \\ \boldsymbol{-1\ se\ v = y \wedge \langle x,v\rangle \in E} \\ \boldsymbol{0\ altrimenti} \end{cases}$$



	(a,b)	(c,a)	(b,c)	(c,d)	(a,d)
a	1	1	0	0	1
b	1	0	1	0	0
c	0	1	1	1	0
d	0	0	0	1	1



	(a,b)	(c,a)	(b,c)	(c,d)	(a,d)
a	1	-1	0	0	1
b	-1	0	1	0	0
c	0	1	-1	1	0
d	0	0	0	-1	-1

Matrici di incidenza – costi operazioni

Nessun particolare vantaggio.

Operazione	**Tempo di esecuzione**
`grado(vertice v)`	$O(m)$
`archiIncidenti(vertice v)`	$O(m)$
`sonoAdiacenti(vertici v e w)`	$O(m)$
`aggiungiVertice(vertice v)`	$O(nm)$
`aggiungiArco(vertici v e w)`	$O(nm)$
`rimuoviVertice(vertice v)`	$O(nm)$
`rimuoviArco(arco e)`	$O(n)$

Cosa devo aver capito fino ad ora

- Quali sono i criteri con cui valutare una rappresentazione di grafi
- Quali sono le rappresentazioni più usate
- Data la descrizione di un grafo e dell'uso che se ne deve fare, sapersi orientare verso la rappresentazione migliore

…se non ho capito qualcosa

- Alzo la mano e chiedo
- Ripasso sul libro
- Chiedo aiuto sul forum
- Chiedo o mando una mail al docente