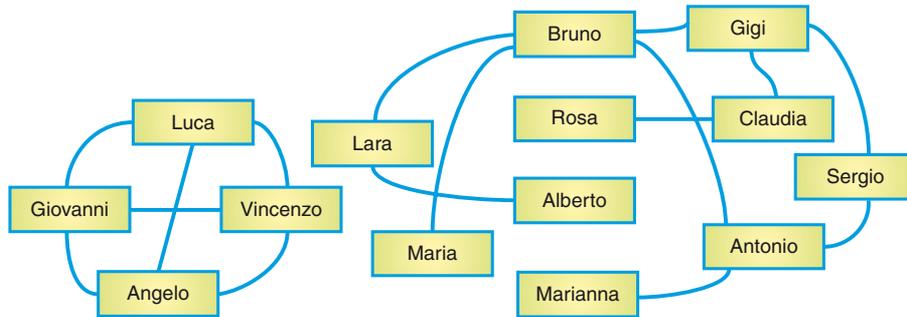


# Grafo e albero

## Il grafo

Le strutture di dati sono di norma tanto più utili quanto più rispettano e riflettono le relazioni tra gli oggetti rappresentati. Definita una relazione (per esempio "essere amico di") e un insieme di elementi oggetto della relazione (per esempio i cittadini di un comune), la rappresentazione grafica della relazione ha un aspetto simile a quello in figura:

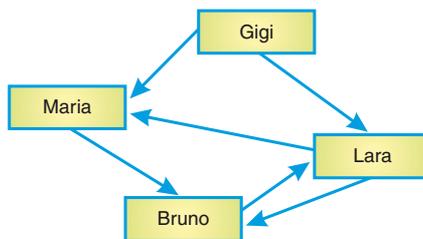


Una tale rappresentazione della relazione viene detta **grafo**.

I **nodi** sono gli elementi del grafo, mentre i segmenti che li uniscono si chiamano **archi**.

Nell'esempio in figura i *nodi* sono i nomi di persona e gli *archi* rappresentano le relazioni di amicizia. La relazione specifica che è stata presa come esempio ha il carattere della **simmetria**: se *Bruno* è amico di *Maria*, allora anche *Maria* è amica di *Bruno*. Questo significa che non è necessario assegnare un verso ai collegamenti relazionali tra gli elementi dell'insieme.

Viceversa, il grafo relativo ad altre relazioni che possono non essere simmetriche (per esempio "ritenere simpatico") necessita di informazioni riguardanti il verso della relazione.



Considerando il grafo rappresentato, si vede per esempio che *Gigi* "ritiene simpatiche" *Maria* e *Lara*, *Lara* a sua volta "ritiene simpatici" *Maria* e *Bruno*. Il grafo di una relazione non biunivoca è detto **grafo orientato**.

Le strutture di dati *pila* e *coda* sono casi particolari di grafi orientati, particolarmente semplici perché ogni elemento è legato con un solo altro elemento dell'insieme su cui sono definite le relazioni. La successione dei nodi che devono essere visitati per raggiungere un nodo prefissato si chiama **cammino** (in inglese **path**).

Si osservi che, nel caso della relazione "essere amico di", l'insieme di relazione si può dividere in due sottoinsiemi non collegati da alcun cammino, il primo composto da *Vincenzo*, *Angelo*, *Luca* e *Giovanni*, e il secondo da tutti gli altri componenti l'insieme.

Tale grafo viene detto **non connesso**, e si contrappone ai grafi **connessi**, nei quali, fissati due nodi qualsiasi, è sempre possibile stabilire un cammino che li congiunge. La proprietà topologica di connessione prescinde dall'eventuale orientamento del grafo, e perciò il secondo grafo presentato è connesso anche se non è possibile seguire un percorso orientato che partendo da *Bruno* arrivi a *Gigi*.

## L'albero

Un **albero** è un **grafo connesso privo di circuiti**, dove per **circuito** si intende un qualsiasi percorso chiuso che permetta, partendo da un elemento, di tornare sul medesimo elemento senza ripercorrere lo stesso tratto due volte.

Anche in questo caso si prescinde dal verso dei collegamenti. Nel secondo grafo della pagina precedente, il percorso *Lara* → *Maria* → *Bruno* → *Lara* è un circuito. Un esempio comune di *albero* è quello che esprime la relazione "essere genitore di" nell'albero genealogico di una famiglia. Altri esempi di albero sono: l'organizzazione del Codice Civile in titoli, capitoli, articoli; la struttura delle responsabilità e delle dipendenze in un'azienda; in generale tutte le situazioni nelle quali è possibile determinare diversi livelli gerarchici.

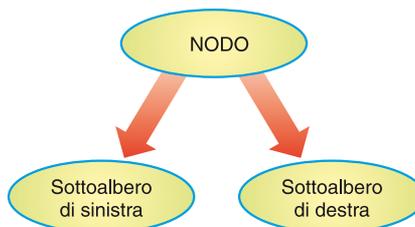
Gli elementi di un insieme di relazione rappresentato da un albero si chiamano **nodi**, mentre con **ramo** si indica un cammino composto da più nodi connessi. Un nodo posto al termine di un ramo è chiamato **foglia**, mentre il nodo dal quale partono i rami è detto **radice**. I nodi sottostanti a una *radice* vengono chiamati nodi **figli**.

Il fatto che l'albero non contenga circuiti garantisce che, partendo dalla radice, si possa raggiungere qualsiasi foglia senza rischiare di finire in un circolo dal quale non si possa uscire.

In informatica, l'organizzazione dei file su disco gestita dal **file system** del sistema operativo, rappresenta un esempio di albero nel quale la *radice* è la directory principale (detta appunto **root**), i *nodi* sono le cartelle e le *foglie* sono i file.

Nella metodologia di **sviluppo top-down** dei problemi, per la scomposizione del problema in sottoproblemi, la struttura ad *albero* è utilizzata per rappresentare la **gerarchia delle funzioni** in un'applicazione informatica.

Un **albero binario** è un particolare albero nel quale da ogni nodo partono al massimo due rami distinti, chiamati sottoalbero sinistro e sottoalbero destro.

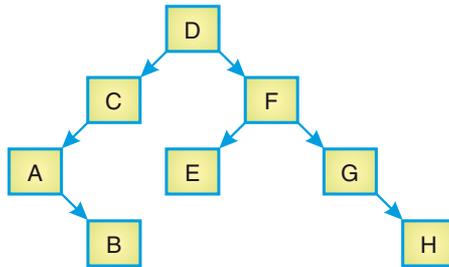


Per ritrovare i dati si possono seguire diversi percorsi sui nodi dell'albero, a seconda di come si vogliono elaborare i dati contenuti nell'albero: si hanno così diversi modi di **visita** o **attraversamento** dell'albero. Quando l'ordine di attraversamento è: sottoalbero di sinistra, radice, sottoalbero di destra, la visita dell'albero si chiama **inordine** o in ordine simmetrico. Altre due modalità di attraversamento dell'albero sono quelle in **preordine** o ordine anticipato e in **postordine** o ordine posticipato. La tabella mostra come le tre modalità differiscano tra loro.

INORDINE	PREORDINE	POSTORDINE
• Sottoalbero sinistro	• Radice	• Sottoalbero sinistro
• Radice	• Sottoalbero sinistro	• Sottoalbero destro
• Sottoalbero destro	• Sottoalbero destro	• Radice

L'attraversamento dei sottoalberi nei tre casi avviene con la stessa modalità dell'attraversamento dell'albero: se l'ordine di attraversamento dell'albero è posticipato, anche l'ordine di attraversamento di ciascun sottoalbero è posticipato. Di seguito vengono descritti i tre modi di visita con l'ordine secondo il quale vengono elaborati i dati nell'albero binario della figura che ha il nodo radice contenente D.

- **INORDINE** i dati vengono elaborati nell'ordine: A B C D E F G H
- **PREORDINE** i dati sono elaborati nell'ordine: D C A B F E G H.
- **POSTORDINE** i dati sono elaborati nell'ordine: B A C E H G F D.



La scelta di attraversare l'albero in un modo o nell'altro dipende ovviamente dal tipo di elaborazione che si vuole eseguire: per esempio, volendo stampare la struttura dell'albero conviene usare il preordine (stampa il dato, poi stampa ciascun sottoalbero).