Introduzione a Grafi e Alberi

Grafo Bipartito	Dati due insiemi, **U** e **V**, una relazione **A** su questi due insiemi è un sottoinsieme di **UxV**. Si può rappresentare **A** sotto forma di grafo bipartito, cioè un grafo nel quale tutti i vertici di una parte sono collegati solo a vertici dell'altra. ...

Alberi Binari

Alberi Binari	Un **albero binario** è tale se ogni nodo ha al più due figli; inoltre si fa sempre distinzione tra il figlio sinistro e destro. I nodi di un albero binario si analizzano tramite un **algoritmo di visita**.
Proprietà	Un **albero binario** si dice **completo** se - tutte le foglie hanno la stessa profondità *h* - tutti i nodi interni hanno 2 figli. Un albero binario si dice **quasi completo** se tutti i livelli, tranne al più l'ultimo, sono completi; nell'ultimo livello possono mancare alcune foglie consecutive a partire dall'ultima foglia a destra. Un albero binario si dice **bilanciato** quando la differenza di altezza tra sottoalbero sinistro e destro è al più di 1, ed i due sottoalberi sono bilanciati.
Algoritmi di Visita (per Alberi Binari)	La **visita di un albero** consiste nel seguire una rotta di viaggio che consenta di esaminare ogni nodo dell'albero esattamente una volta. I più comuni **algoritmi di visita** sono tre: 1. **Visita in Pre-Ordine** Si applica ad un albero non vuoto ed effettua prima l'analisi della radice dell'albero e poi la visita dei due sottoalberi (prima il sinistro e poi il destro), effettuata con lo stesso metodo della radice. 2. **Vista in Post-Ordine** Si applica ad un albero non vuoto ed effettua prima la visita dei sottoalberi, prima il sinistro e poi il destro, ed in seguito l'analisi della radice dell'albero. 3. **Visita Simmetrica** Effettua prima la visita del sottoalbero sinistro, poi l'analisi della radice ed infine la visita del sottoalbero destro.
Rappresentazioni: Vettore	Tra le possibili **rappresentazioni** troviamo quella con **Vettore**. E una rappresentazione sequenziale dove abbiamo: - radice in prima posizione;

	- generico nodo **p** in posizione **i**; - figlio sinistro in posizione **2*i**; - figlio destro in posizione **2*i+1** Questa rappresentazione presenta alcune problematiche, come il fatto che alcune posizioni del vettore non corrispondono ad alcun nodo dell'albero. Per risolvere questo problema si può usare un vettore booleano, nel quale le posizioni occupate sono identificate con *true*. Ad ogni modo, questa rappresentazione comporta uno **spreco di spazio** e anche il tipico limite sul numero di elementi e difficoltà di spostamento degli array.
Rappresentazioni: Rappresentazione Collegata	Ogni valore **T** di tipo albero binario può anche essere rappresentato con una **lista**. Abbiamo che se **T** è vuoto, allora anche la lista sarà vuota. Se invece **T** non è vuoto, la lista avrà 3 elementi: 1. il primo rappresenta la radice di **T**; 2. il secondo rappresenta, nello stesso modo, il sottoalbero sinistro; 3. il terzo rappresenta il sottoalbero destro. Si può anche utilizzare una rappresentazione con parentesi: - ( ) albero vuoto - (a) albero costituito solo dalla radice - (a ( ) ( )) albero costituito da radice e figlio sinistro/destro vuoto Per quanto riguarda la **l'implementazione** della rappresentazione collegata, ci sono diverse possibilità: - **Matrice** Si utilizza una matrice dove sulle colonne troviamo sottoalbero sinistro, nodo e sottoalbero destro; se il figlio del nodo nella colonna centrale esiste, il valore corrisponderà alla sua posizione, altrimenti sarà 0. - **Puntatori** Si usa una lista (di liste) dove abbiamo 4 campi: uno che punta al sottoalbero sinistro uno per il nodo dal quale nascono i sottoalberi uno che punta al sottoalbero destro uno che punta al nodo padre del sottoalbero corrente
Esempio di Rappresentazione con Puntatori