

Corso di Laurea in Informatica, A.A. 2024-2025.

Laboratorio di Algoritmi e Strutture Dati

Esame finale

April 22, 2025

## AlberoVP

Sia  $T$  un albero con le seguenti caratteristiche:

1. Ogni nodo contiene una sola informazione di cui non sappiamo il tipo ma tutte le informazioni contenute nei nodi sono dello stesso tipo. Alternativamente si può scegliere che le informazioni dei nodi siano stringhe o numeri interi.
2. Non è noto il numero massimo di figli che ogni nodo può avere.
3. L'inserimento di un nodo nell'albero (esclusa la radice) è possibile solo specificando il padre del nodo che verrà inserito e l'informazione che esso conterrà.
4. Quando si inserisce un nodo  $v$  come figlio di un altro nodo  $u$  già presente nell'albero, il nodo  $v$  diviene l'ultimo figlio di  $u$ .

Rappresentare l'albero  $T$  mediante due strutture indicizzate (array): una memorizza i nodi dell'albero, l'altra invece memorizza il padre di ogni nodo. La corrispondenza fra padre e figlio è realizzata in modo che il nodo che occupa la posizione  $i$ -esima della seconda struttura è il padre del nodo che occupa la posizione  $i$ -esima della prima struttura. Ad esempio, l'albero in Figura 1 può essere rappresentato mediante la struttura

$\{Claudia, Marco, Luca, Giulia, Silvia, Ugo, Andrea, Gianna, Carlo\}$

per la memorizzazione dei nodi e da

$\{null, Claudia, Claudia, Claudia, Marco, Marco, Giulia, Giulia, Andrea\}$

per la memorizzazione dei padri.

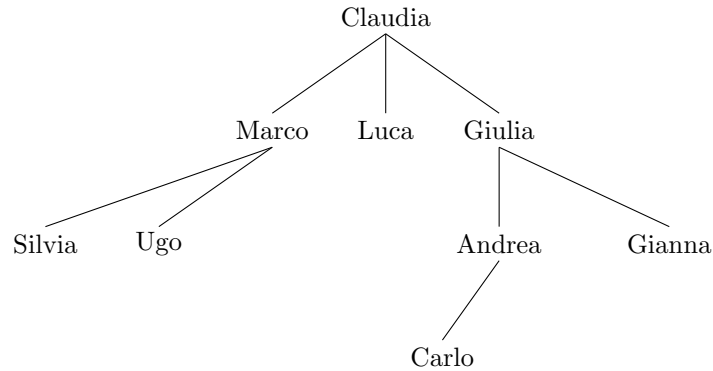


Figure 1: Un albero

Si assuma che l'inserimento di un nodo  $u$  come figlio di un nodo  $v$  avvenga in modo tale che siano coinvolte le ultime posizioni delle strutture. Ad esempio se venisse inserito un nuovo figlio *Cosimo* del nodo *Marco*, le due strutture relative all'esempio precedente diventano:

$\{Claudia, Marco, Luca, Giulia, Silvia, Ugo, Andrea, Gianna, Carlo, Cosimo\}$

per la memorizzazione dei nodi e da

$\{null, Claudia, Claudia, Claudia, Marco, Marco, Giulia, Giulia, Andrea, Marco\}$

per la memorizzazione dei padri. E' chiaro quindi che uno stesso albero può avere rappresentazioni diverse secondo l'ordine di arrivo dei nodi con cui si costruisce l'albero: un'altra possibilità di rappresentazione dell'albero precedente è data dalle strutture

$\{Claudia, Marco, Silvia, Luca, Giulia, Andrea, Gianna, Carlo, Ugo, Cosimo\}$

per la rappresentazione dei nodi e

$\{null, Claudia, Marco, Claudia, Claudia, Giulia, Giulia, Andrea, Marco, Marco\}$

per la rappresentazione dei padri.

Dopo aver costruito una classe `NodoVP` nella maniera opportuna, utilizzarla per l'implementazione di una classe `AlberoVP` che realizzi le caratteristiche della rappresentazione sopra descritta. Dovranno essere implementati i metodi per le seguenti operazioni sugli alberi:

- inserire la radice in un albero vuoto e restituirla;
- inserire una nuova radice in un albero non vuoto in modo che la vecchia radice diventi figlia di quella nuova, che deve essere restituita;
- restituire la radice di un albero;
- restituire la lista dei figli di un nodo;
- restituire il padre di un nodo;
- restituire il numero di nodi dell'albero;
- restituire il livello di un nodo;
- aggiungere e restituire un nodo con informazione data come figlio di un nodo già presente (che sarà passato come parametro al metodo, insieme all'informazione);
- restituire il numero di figli di un nodo dell'albero;
- restituire il contenuto di un nodo dell'albero;
- restituire l'altezza dell'albero;
- restituire il numero di figli di un nodo;
- restituire il numero delle foglie di un nodo;
- cambiare il contenuto di un nodo dell'albero;
- restituire la lista dei nodi dell'albero visitato in profondità;
- restituire la lista dei nodi dell'albero visitato in ampiezza;
- restituire una stringa che rappresenti l'albero: riferendosi alla Figura 1 la stringa deve avere il seguente formato:

*Claudia[Marco[Silvia[ ], Ugo[ ] ], Luca[ ], Giulia[Andrea[Carlo[ ]], Gianna[ ] ] ]*

Deve poter essere possibile visualizzare i vari parametri di un albero o di un nodo, quali la lista delle informazioni dei figli di un nodo, l'informazione di un nodo, la lista delle informazioni dei nodi derivante da una certa visita, l'altezza dell'albero, etc... Per questo può essere utile implementare dei metodi appositi che ne facilitino la stampa.

Fornire, se necessario, un breve ma chiaro commento ai metodi implementati, così come una breve descrizione delle variabili utilizzate nelle due classi.

Se ritenuto opportuno, fornire un metodo `main run` in cui siano testati i metodi implementati.

## Istruzioni per la consegna

Il progetto può essere realizzato anche a piccoli gruppi di al più tre autori. Riportare all'interno del codice, all'inizio di ogni classe come commento, Nome, Cognome e Matricola di ogni autore, in maniera chiara.

Solo uno degli autori deve consegnare il progetto: colui che ha il cognome che appare prima nell'ordine alfabetico. La consegna avviene tramite la piattaforma e-learning del corso. Il termine ultimo per la consegna è 5 giorni prima della data dello scritto dell'appello in cui si intende discutere il progetto. Deve essere consegnata la cartella del progetto presente nello workspace di Eclipse. La cartella va nominata con i cognomi degli autori in ordine alfabetico (Es: Bianchi\_Rossi\_Verdi).

Accertarsi di consegnare NON in modalità bozza. In tal caso infatti non arriverebbe l'email di avvertimento di avvenuta consegna. E' necessario cliccare sul pulsante "Consegna Compito". Al momento della prenotazione all'esame, nel caso in cui il progetto sia stato svolto in gruppo, segnalare cortesemente chi è l'autore che ha consegnato.