Università degli Studi dell'Aquila

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Università degli Studi dell'Aquila

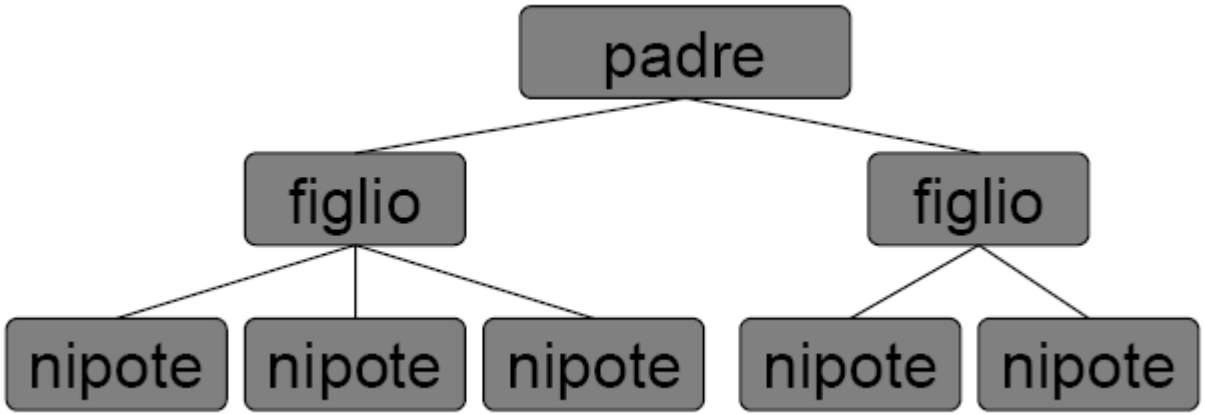
Corso di Algoritmi e Strutture Dati con Laboratorio

BINARY TREE

Albero

- Un albero è una collezione di elementi (nodi), sui quali è definita una relazione di discendenza con due proprietà:

1. esiste un solo nodo radice senza predecessori
2. ogni altro nodo ha un unico predecessore



Albero

- Un albero è una coppia $T =< V, E >$ dove V e un insieme di nodi (o vertici) , $E \subseteq V \times V$ è una relazione su V
- un nodo (o vertice) $v \in V$ è un oggetto a cui può essere associata informazione
- un arco $e \in E$ è una connessione tra nodi:

$$e =< v_{padre}, v_{figlio} >$$

Proprietà: esiste esattamente un cammino tra ogni coppia di nodi

- si dice cammino una sequenza di nodi distinti in cui i nodi successivi sono connessi da un arco

Motivazioni

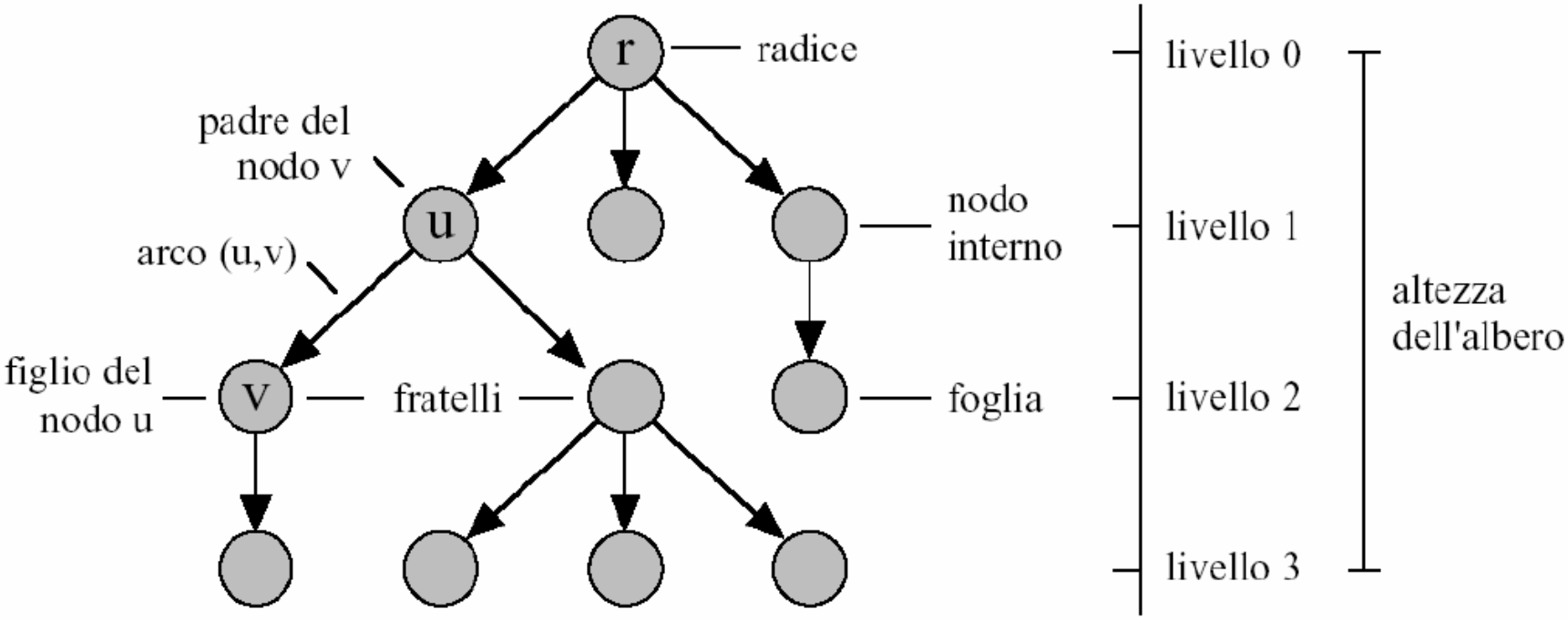
Gli alberi giocano un ruolo centrale nella progettazione e nell'analisi degli algoritmi e nelle applicazioni informatiche:

- impieghiamo gli alberi per descrivere le proprietà dinamiche degli algoritmi
- gli alberi sono strettamente legati all'analisi sintattica dei linguaggi di programmazione
- Un file system ha una organizzazione ad albero ... Usiamo frequentemente strutture che rappresentano implementazioni concrete di alberi

Terminologia

- **Livello/Profondità** di un nodo: distanza di un nodo dalla radice; la radice ha livello 0; ogni nodo interno (ovvero ≠ radice) ha livello pari a quello del padre + 1
- **Altezza** di un albero: massimo fra i livelli di tutti i suoi nodi

Terminologia



Terminologia

- **Antenati** di un nodo: nodi raggiungibili salendo di padre in padre lungo il cammino verso la radice
- **Discendenti** di un nodo: nodi raggiungibili scendendo di figlio in figlio nell'albero
- **Albero ordinato**: albero in cui è specificato un ordine totale tra i figli di ciascun nodo
- **Grado** (di uscita) di un nodo: numero dei figli del nodo
- **Albero d-ario**: albero i cui nodi hanno grado $\leq$ d
- **Albero d-ario completo di altezza h**: albero in cui tutte le foglie hanno profondità h ed ogni altro vertice ha grado =d

Definizioni

- Un **albero ordinato d-ario** è vuoto oppure consiste di una radice connessa ad una sequenza ordinata di alberi ordinati d-ari
  - ogni nodo è la radice di un sotto-albero formato dal nodo stesso e da tutti i suoi discendenti
- Un **albero binario** è un albero ordinato 2-ario: ogni nodo ha esattamente un padre e (al più) due figli ordinati detti figlio sinistro e figlio destro

Proprietà

- Il livello i-mo di un albero binario contiene al più 2^i nodi
- Un albero binario completo di altezza h contiene $2^{h+1} - 1$ nodi (progressione geometrica: $2^0 + 2^1 + 2^2 + \ldots + 2^h$)
- Un albero binario completo di N nodi ha un'altezza $h = \log_2 (N + 1) - 1$
- L'altezza di un albero binario di N nodi è compresa tra
  - $h = N - 1$
  - $h = \lceil \log_2 (N + 1) \rceil - 1$

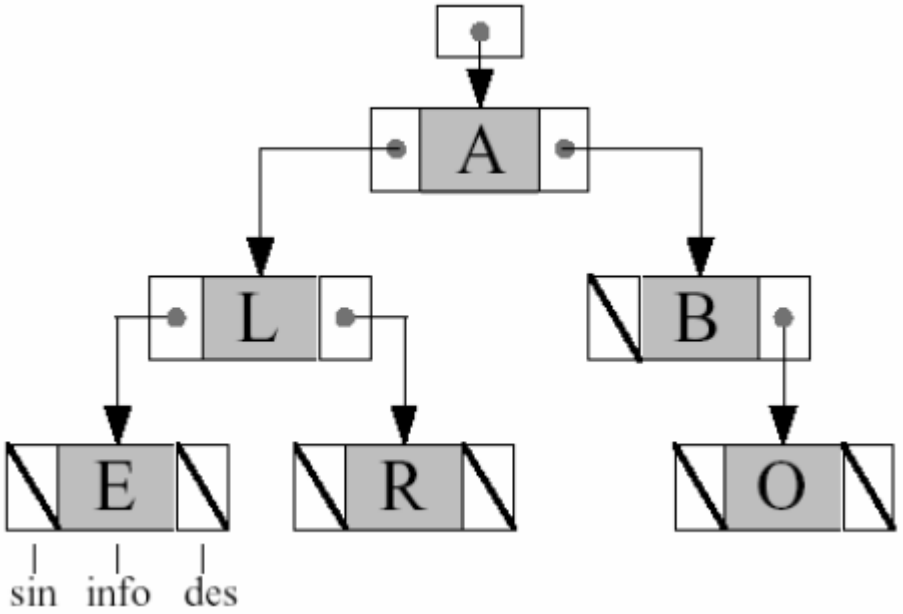
ADT Btree[Item]

1. Insieme di domini: { Btree, Item, Boolean }
2. Operazioni:
   - `newBtree()`$\rightarrow$`Btree` (crea un nuovo albero binario vuoto)
   - `left(Btree)`$\rightarrow$`Btree` (individua il sotto-albero sinistro)
   - `right(Btree)`$\rightarrow$`Btree` (individua il sotto-albero destro)
   - `data(Btree)`$\rightarrow$`Item` (recupera l'elemento contenuto nella radice)
   - `make(Btree,Item,Btree)`$\rightarrow$`Btree`
     (costruisce un nuovo albero a partire dai tre parametri)
   - isEmptyBtree(Btree)$\rightarrow$Boolean
     (verifica se un albero binario è vuoto o meno)
3. Assiomi: $\forall i \in \mathit{Item}, \forall L, R \in \mathit{Btree}$:

```
isEmptyBtree(newBtree())=TRUE
isEmptyBtree(make(L,i,R))=FALSE
left(make(L,i,R))=L
right(make(L,i,R))=R
data(make(L,i,R))=i
```

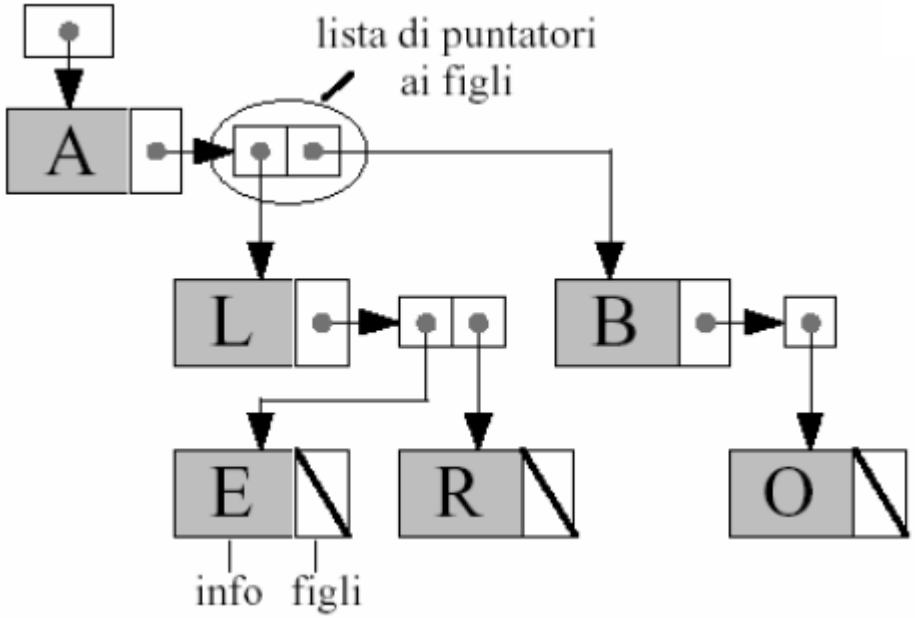
Rappresentazioni collegate

- Rappresentazione con **puntatori ai figli sinistro e destro**
- generalizzabile ad alberi ordinati d-ari



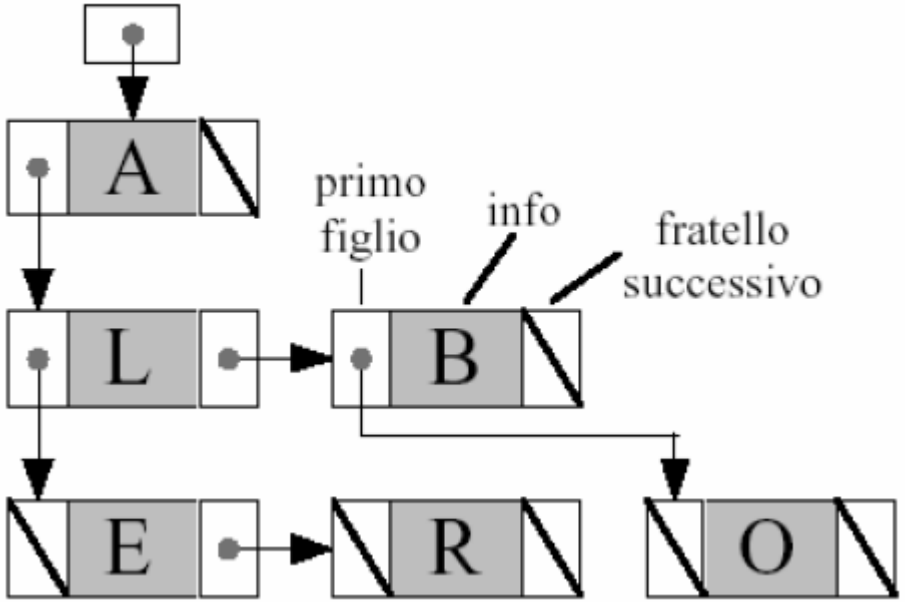
Rappresentazioni collegate

- Rappresentazione con **liste di puntatori ai figli**
- generalizzabile ad alberi ordinati d-ari e con numero illimitato di figli



Rappresentazioni collegate

- Rappresentazione di tipo **primo figlio-fratello successivo**
- generalizzabile ad alberi ordinati con numero arbitrario di figli



Rappresentazione mediante array

- Un albero binario di altezza h può essere rappresentato mediante un array di dimensione pari al numero di nodi di un albero completo, ossia $N = 2^{h+1} - 1$