

# Albero

lunedì 6 giugno 2022 18:27

- Grafo non orientato, connesso ed aciclico
- Albero binario = Per ogni nodo abbiamo al più 2 nodi collegati
- Antenato : prima nel cammino
- Discendente: dopo nel cammino
- Sottoalbero: albero sotto un determinato nodo
- Parent: padre del nodo
- Grado: Numero di figli che un nodo ha
- Foglia: Nodo con grado 0
- Profondità: quanto lontano è dalla radice (parent)
- Altezza albero: massima profondità
- Completo: Ogni foglia ha la stessa profondità e tutti i nodi non foglia hanno 2 figli  
-> altezza =  $\log_2 n$
- Root[T] -> punta al parent, la radice
- Ogni nodo ha 3 puntatori: left, right, parent  
Foglie: left, right = null  
Root: parent = null
- Successore: Il minimo numero più grande di un determinato nodo  
-> Minimo sotto albero di destra

Un albero binario particolare è quello di ricerca:

- Albero binario dove, per ogni nodo abbiamo sempre a sinistra Un valore più piccolo e a destra quello più grande

Inserimento:

- Per fare l'inserimento, bisogna seguire questo:  
Parti dalla radice. E' il nostro valore  $k >$  della radice?  
Se si allora andiamo verso destra, in caso contrario Andiamo verso sinistra.  
Continuiamo così fino a che non troviamo un valore null

Eliminazione:

- Qui abbiamo 2 casi:
  - o Cancellare un nodo con 1 figlio  
E' il caso più Semplice. Cancelliamo il nodo  
Ed uniamo il padre del nodo cancellato con L'unico figlio. L'albero rimane uguale
  - o Cancellare un nodo con 2 figli  
In questo caso, bisogna mettere il successore di quel nodo

Ricerca:

SBT\_Search(T, k):

```
If x == null:
    Return x
Else:
    If x.key == k:
        Return x
    Else:
        If x.key > k:
            R = SBT_Search(x.left, k)
            Return r
        Else:
            R = SBT_Search(x.right, k)
            Return r
```

Se l'albero è bilanciato,  $O(\log n)$

Senò,  $O(h)$

SBT\_min(X):

```
If x == null
    Return x
While x.left != null
    X = x.left
Return x
```

AVL -> Quanto un albero è sbilanciato

- Per ogni nodo si inserisce un valore che ci dice quanto è sbilanciato



-> Il nodo è bilanciato



-> Sbilanciato a sinistra



-> Doppia-  
mente  
Sbilanciato a  
sinistra



-> Sbilanciato solo di 1

\-> Questo si può ruotare e verrebbe bilanciato