Alberi e BST

Cos’ è un albero?

Un albero è un grafo diretto in cui ogni nodo può avere un solo arco entrante ed un qualunque numero di archi uscenti

* **Se un nodo non ha archi uscenti si dice "foglia"**
* **Se un nodo non ha archi entranti si dice "radice"**

In un albero non ci sono nodi con due o più archi entranti, per ogni albero vi deve essere una ed una sola radice.

Immagine che contiene testo, diagramma, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene schermata, diagramma, cerchio, testo

Descrizione generata automaticamente

**Un albero deve aciclico cioè senza cicli**

Gli albero sono generazioni delle liste

– Ogni elemento ha più di un solo successore.

– Utili per rappresentare partizioni ricorsive di insiemi e strutture gerarchiche

**Alberi come strutture ricorsive**

* Il nodo da cui un arco parte si dice **padre**.
* Un nodo a cui questo arriva si dice **figlio.**
* Due nodi con lo stesso padre sono detti **fratelli.**
* da ogni nodo non-foglia di un albero si dirama un sottoalbero, quindi si intuisce la natura ricorsiva di questa struttura dati.

Immagine che contiene testo, diagramma, Carattere, linea

Descrizione generata automaticamente

**Definizione ricorsiva: un albero è un insieme di nodi che**

* è vuoto
* ha un nodo denominato radice da cui discendono zero o più sottoalberi che sono essi stessi alberi
* dato un nodo, i nodi che appartengono al suo sottoalbero si dicono suoi discendenti
* dato un qualunque nodo, i nodi che si trovano nel cammino dalla radice

ad esso sono i suoi ascendenti (per esempio, B ed A sono ascendenti di C)

**Cammini e Livelli**

* **il livello di un nodo è la sua distanza dalla radice**
* **la radice ha livello 0, i suoi figli livello 1, i suoi nipoti livello 2 e così via**
* **i fratelli hanno lo stesso livello ma non tutti i nodi dello stesso livello sono fratelli**
* **La profondità di un albero è la lunghezza del cammino più lungo dalla**

**radice ad una foglia**

**Immagine che contiene linea, diagramma

Descrizione generata automaticamente**

**La profondità di un albero definito ricorsivamente**

**La radice ha profondità 0 i figli (della radice) profondità 1,**

**etc.**

**Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente**

**Albero bilanciato**

Un albero di profondità h è bilanciato se:

1. **Tutte le foglie si trovano allo stesso livello.**

**2. Dato k numero massimo di figli per nodo, ogni nodo interno (inclusa la radice) ha esattamente k figli.**

**Immagine che contiene linea, diagramma, modello

Descrizione generata automaticamente**

**Albero binario di ricerca (BST (Bynary Search Tree))Immagine che contiene schizzo, diagramma, bianco, Line art

Descrizione generata automaticamente**

La penultima immagine non rappresenta un albero binario di ricerca ma una lista e, pertanto si perdono tutti i vantaggi del BST.

Ad ogni livello n un albero binario può contenere (al più) 2n nodi

Il numero totale di nodi di un albero (incluse le foglie) di profondità n

è al massimo 2(n+1)-1

**Implementazione di un albero**

**Immagine che contiene diagramma, cerchio, bianco, schizzo

Descrizione generata automaticamenteImmagine che contiene diagramma, Piano, Disegno tecnico, schizzo

Descrizione generata automaticamente**

**Ogni nodo dell’albero ha i seguenti attributi:**

* **Nodo SX**
* **Parent**
* **Dato**
* **Nodo DX**

**I Bynary Search Tree**

In un albero binario di ricerca si ha la radice o root e poi gli altri nodi e come nodi terminali le foglie. Nell’albero binario sia dato un nodo generico X e uno Y. Se Y un nodo nel sottoalbero sinistro di radice x allora y.valore < x.valore. Altrimenti y.valore ≥ x.valore. **Ciò significa che a SX si hanno i nodi più piccoli, invece, a SX si hanno i nodi piu Grandi.**

Immagine che contiene schizzo, cerchio, bianco, arte

Descrizione generata automaticamente

**Con i BST si possono eseguire svariate operazioni:**

* **Inserimento di un nuovo nodo**
* **Ricerca di un nodo nell’ BST**
* **MAX**
* **MIN**
* **Cancellazione di un nodo**

**Immagine che contiene schizzo, appendiabiti, arte, design

Descrizione generata automaticamente con attendibilità media**

**Visita di un BST**

**Si hanno 3 algoritmi:**

1. **La visita preorder** visita prima la radice, quindi il sottoalbero sinistro e da ultimo quello destro
2. **La visita inorder** processa prima il sottoalbero sinistro, quindi la radice ed infine il sottoalbero destro
3. **La visita postorder** processa prima il sottoalbero sinistro, poi quello destro ed infine la radice

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Immagine che contiene testo, schermata, diagramma, bianco

Descrizione generata automaticamenteImmagine che contiene testo, Carattere, schermata, algebra

Descrizione generata automaticamente

**Costi delle varie operazioni**

* **Complessità: T(n)= Θ(n) (visita di un BST )**
* **Complessità: T(n)=O(h) con H profondità dell’albero.**
* **Complessità: T(n)=O(h) ricerca del massimo e del minimo.**

**Successore di X (con X nodo generico)**

1. Il sottoalbero destro di x è non vuoto: caso banale
2. Il sottoalbero destro di x è vuoto

* Minimo antenato di x il cui figlio sinistro è anche un antenato di x
* Ogni nodo è antenato di se stesso

**Cancellazione di un no dal BST**

Si hanno tre casi da considerare:

1. Z non ha figli la procedura è banale: eliminiamo z emodifichiamo il padre in modo che il puntatore a z diventi un puntatore a NULL.
2. Z ha un solo figlioImmagine che contiene schizzo, diagramma, disegno

   Descrizione generata automaticamente
3. Z ha due figli

* Cerchiamo il successore y di z
* Si troverà certamente nel sottoalbero destro di radice z
* y prende la posizione di z nell’albero
* La rimanente parte del sottoalbero destro (di z) diventa il nuovo albero destro di y