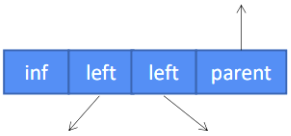
Alberi binari di ricerca

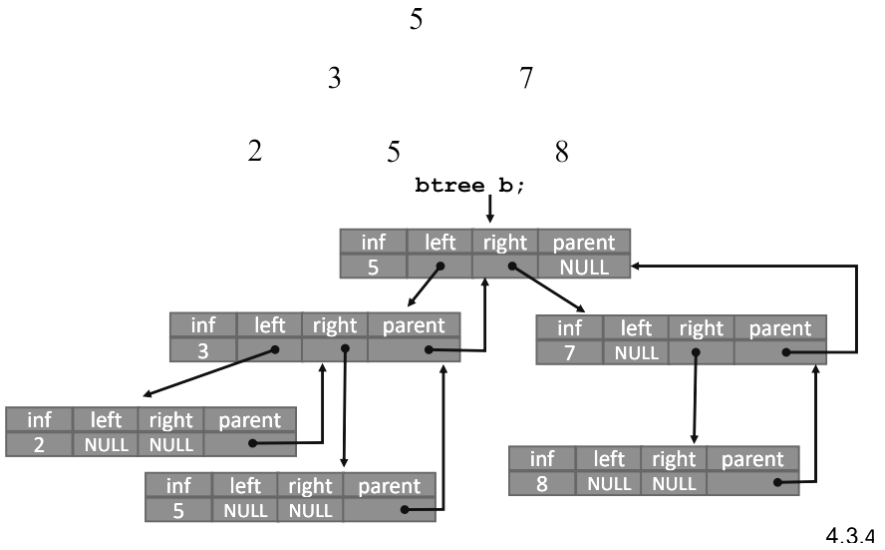
Albero binario: albero dove il numero di figli per nodo è al massimo due facendo distinzione tra figlio dx e sx.

```
//definizione della struttura nodo
struct bnode {
      tipo_inf inf;
      bnode* left;
      bnode* right;
      bnode* parent;
   };

//dichiarazione del tipo di dato binary tree
typedef bnode* btree;
```



Rappresentazione di un albero binario



4.3.4

Alberi di ricerca

Strutture dati dinamiche ad albero usate per localizzare valori di chiave in insiemi di elementi (chiave, valore) dove esiste un ordinamento totale sulle chiavi. Possono essere utilizzati per implementare dizionari o code di priorità.

Albero binario di ricerca: albero binario (BST - binary search BST) è un albero con le seguenti proprietà:

- Ogni **nodo n** ha
  - un contenuto informativo **value(n)**
  - una chiave **key(n)** presa da un dominio totalmente ordinato (ovvero su cui è definita una relazione d'ordine totale <)
- Sia n' un nodo nel **sottoalbero sinistro** di n allora *key(n')≤key(n) (oppure key(n')<key(n))*
- Sia n' un nodo nel **sottoalbero destro** di n allora key(n')> key(n) (oppure key(n') ≥ key(n))

Implementazione

8 ->12

`void bst_insert(bst&, bnode*)`
L'inserimento non è un semplice aggancio di puntatori ma uno scorrere l'albero per trovare la posizione corretta per inserire il nodo affinché venga mantenuta la proprietà di ricerca. Il nodo viene sempre aggiunto come figlio di un nodo foglia.

`Bst` viene passato per riferimento perché all'inizio, alla prima scansione, si ha la necessità che venga inserita una radice

Si parte dalla radice e confronto la chiave tra il nodo corrente z e il nodo da inserire n.

- Se *key(n) < key(z)* la scansione prosegue nel sottoalbero sinistro
- Se *key(z) < key(n)* la scansione prosegue nel sottoalbero di destra

La scansione termina quando il sottoalbero selezionato è vuoto, ossia ha valore NULL (non ci sono altri nodi dell'albero); a questo punto si può inserire il nuovo nodo.

Relazione: tutto ciò che sta a sinistra di un certo nodo è minore; tutto ciò che sta a destra è maggiore.

Esempio inserimento valore 13