**BINARY SEARCH TREE**

**(Cây Nhị Phân Tìm Kiếm)**

- Cây nhị phân tìm kiếm là cây nhị phân có:

+ Một node bất kì luôn thỏa điều kiện node con bên trái có giá trị nhỏ hơn node cha và node con bên phải có giá trị lớn hơn node cha.

+ Giá trị các khóa của cây con phải lớn hơn giá trị node gốc.

+ Giá trị các khóa của cây con trái nhỏ hơn giá trị của node gốc.

+ Cây con bên trái và cây con bên phải đều phải là cây nhị phân tìm kiếm.

- Khi thêm một khóa vào cây thì nó phải là lá

***Hình A***

Diagram

Description automatically generated

Diagram

Description automatically generated***Hình B***

***Hình C***

A picture containing text, whiteboard

Description automatically generated

- Function prototype (Nguyên mẫu hàm hay hàm nguyên mẫu) về insert

int insertTree(Node\*& root,int key)

{

if(root==NULL)

{

root=createNode(key);

}

else if(root->value > key)

{

insertTree(root->pLeft,key);

}

else if(root->value < key)

{

insertTree(root->pRight,key);

}

else return 0;

return 1;

}

- Xoay trái tại A:

A thành con phải của A (là B)

Con trái của B thành con phải của A

Diagram

Description automatically generated

Diagram

Description automatically generated

- Xoay phải tại B:

B thành con trái của B (là A)

Con phải của A thành con trái của B

Diagram

Description automatically generated with medium confidence

Diagram

Description automatically generated

- Xóa Node bất kì, có 3 trường hợp:

+ Node đó là node lá: Xóa bình thường

+ Node đó có một node con: Nối node ông xuống node cháu, xóa node cha

+ Node đó có hai node con: Tìm node có giá trị lớn nhất ở cây con bên trái hoặc node có giá trị nhỏ nhất ở cây con bên phải, sau đó đổi vị trí node cần xóa với node vừa tìm được, lúc này đưa về bài toán xóa node lá hay xóa node chỉ có một node con.