Arbre binaire de recherche

Recherche un élément dans l'abre

Si la branche gauche n'est pas null et que la clé est plus petite ou égale que la branche gauche on relance dans ce coté. Sinon, si la branche droite n'est pas null et que la clé est plus petite ou égale que la branche droite on relance dans ce coté

```
public static boolean search(int key, BTree<Integer> ABR){
    if(ABR.data() == key){
        return true;
    }
    if (ABR.left() != null && key <= ABR.left().data()){
        return search(key, ABR.left());
    }else if(ABR.right() != null && key <= ABR.right().data()){
        return search(key, ABR.right());
    }
    return false;
}
```

Insert un élément dans l'arbre

```
public static void insert(int key, BTree<Integer> ABR){
    if(ABR.data() == null){
        ABR = new BTree<>(key);
    }

    //On compare la valeur à inserer avec la donnée du noeud actuel, si
elles sont vides alors on peut inserer
    if(key <= ABR.data() && ABR.left() == null){
        ABR.setLeft(new BTree<>(key));
        return;
    }else if(key >= ABR.data() && ABR.right() == null){
        ABR.setRight(new BTree<>(key));
        return;
    }

    //Si les branches ne sont pas vide, alors on relance la fonction sur
le sous-arbres
    if (ABR.left() != null && key <= ABR.data()){
        insert(key, ABR.left());
        return;
    }else if(ABR.right() != null && key >= ABR.data()){
        insert(key, ABR.right());
        return ;
    }
```

```
        return;
    }
```

Verifie si un arbre est un arbre binaire de recherche

```
public static boolean isBST(BTree<Integer> btree){
    boolean isBST = true;

    if(btree == null){
        return false;
    }
    //On verifie si la gauche est null, si elle ne l'est pas on verifie la
condition, si la condition est respecté on relance, sinon faux.
    if(btree.left() != null){
        if(btree.data() >= btree.left().data()){
            isBST = isBST(btree.left());
        }else{
            return false;
        }
    }
    //On verifie si la droite est null, si elle ne l'est pas on verifie la
condition, si la condition est respecté on relance, sinon faux.
    if(isBST && btree.right() != null){
        if(btree.data() <= btree.right().data()){
            isBST = isBST(btree.right());
        }else{
            return false;
        }
    }
    return isBST;
}
```