AED teorica
10/12/12

② Private BSTNode<K,V> BSTCreate
(BSTNode<K,V> currentRoot){

```
BSTNode<K,V> currentNode = NULL;
if(currentRoot != NULL)
  currentNode = new BSTNode<K,V>(currentroot
    .getKey(), currentroot.getvalue(),
    bstCreate(currentroot.getLeft()),
    bstCreate(currentroot.getRight());

return currentNod
```

// constructor que chama BstCreate

```
Public BinarySearchTree(BinarySearchTree
                        <K,V> otherTree){
  if(otherTree != null){
    root = bstCreate(otherTree.root);
    this.currentSize = otherTree.size();
```

③

```
Public class FusionIterator <K extends comparator<K>>
                implements Iterator<K>{

    Private Iterator <K> it1;
    Private Iterator <K> it2;

    Private K next1,
    Private K next2;


Private K nextONe 'Iterator <K> it){
    if(it.hasNext())
        return it.next();
    else
        return null;

Public void rewind(){
    it1 rewind();
    it2 rewind();
    next1 = nextOne (it1);
    next2 = nextOne (it2);
}

Public boolean hasnext(){
    return(next1 != NULL)||(next2 != null);
}
```

Se não tem próximo
lançar exPeção

```
else
    next1 ≠ Null
        next2 ≠ null
            next1.compareTo(next2)<=0
                -> avança next1
            else
                avança next2

        else
            avança next1
    else
        avanço next2
```

④

TDA< String, Integer > Socios

↑ Socios ↑ Ano filiação

long n_Votos
int anoCorrente

currentYear()

↳ devolve a variável anoCorrente O(1)