1.

* (h+1) nodes
* (2h+1) nodes
* (2^h) nodes

2.

public static <T> boolean isomorphic(BTNode<T> T1, BTNode<T> T2) {

if ((T1!=null && T2==null) || (T1==null && T2!=null) {

return false;

}

if (T1==null && T2==null) {

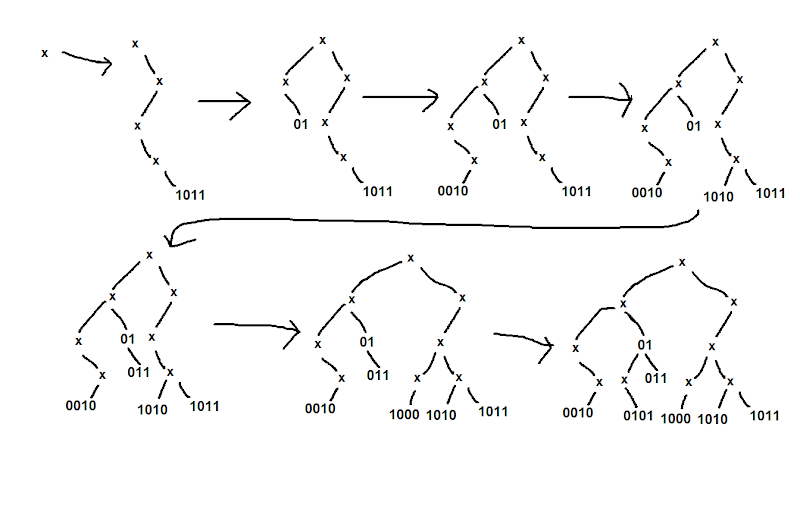
return true;

}

return (isomorphic(T1.left,T2.left) && isomorphic(T1.right, T2.right));

}

3.



* 1011: 4 units of time

01: 2 units of time

0010: 4 units of time

1010: 4 units of time

011: 3 units of time

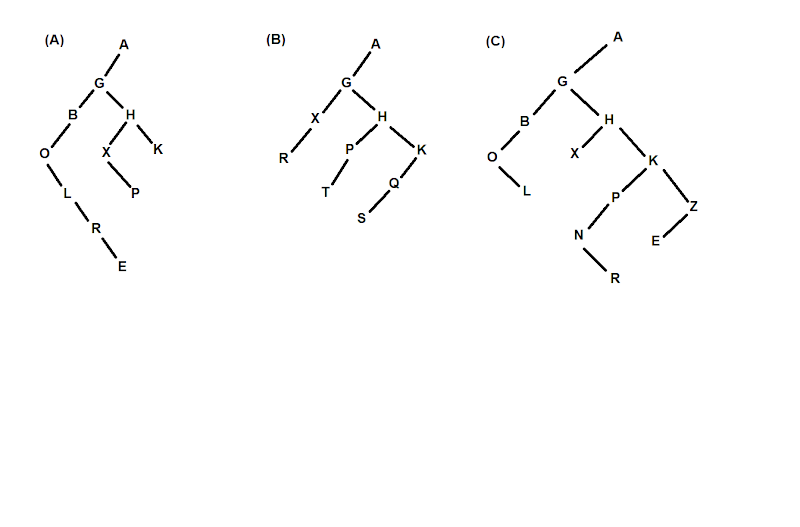
1000: 4 units of time

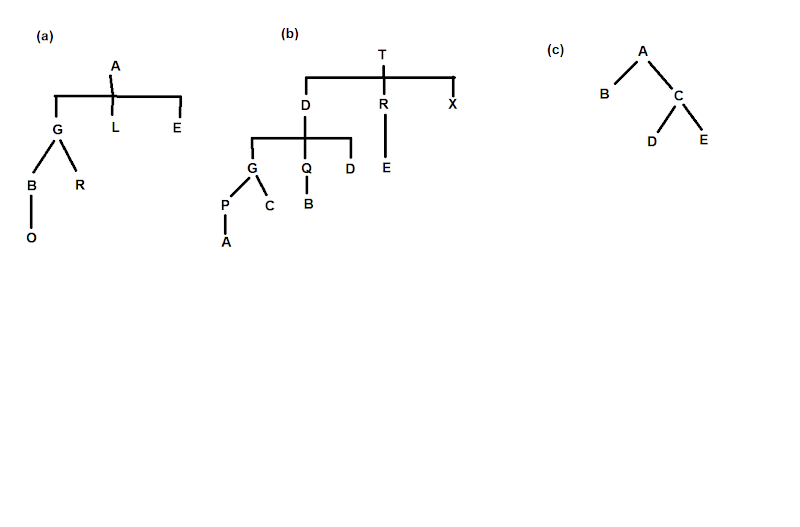
0101: 4 units of time

It would take 25 total units of time to build the tree.

* 15 units of time
* k units of time
* n units of time

4.





5.

public static <T> BTNode<T> genTreeParent(BTNode<T> x) {

if (x.parent.left == x) {

return x.parent;

} else {

return genTreeParent(x.parent);

}

}

public static <T> BTNode<T> genTreekthChild(BTNode<T> x, int k) {

if (x.left == null || k == 0) {

throw new NoSuchElementException();

}

int count = 0;

BTNode<T> kid = x.left;

while (kid != null) {

count++;

if (count == k)

return kid;

kid = kid.right;

}

throw new NoSuchElementException();

}