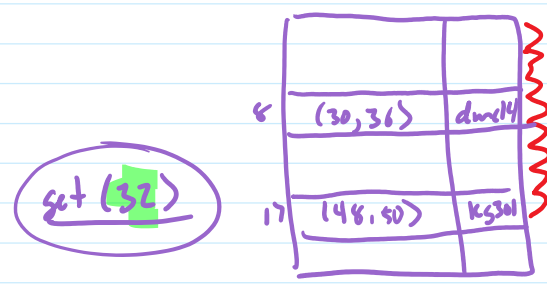
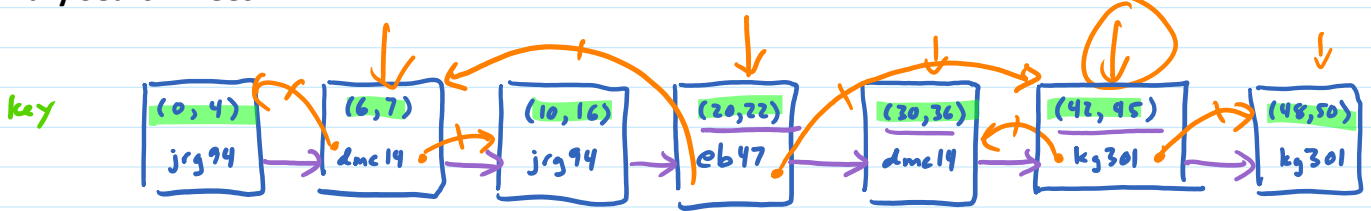
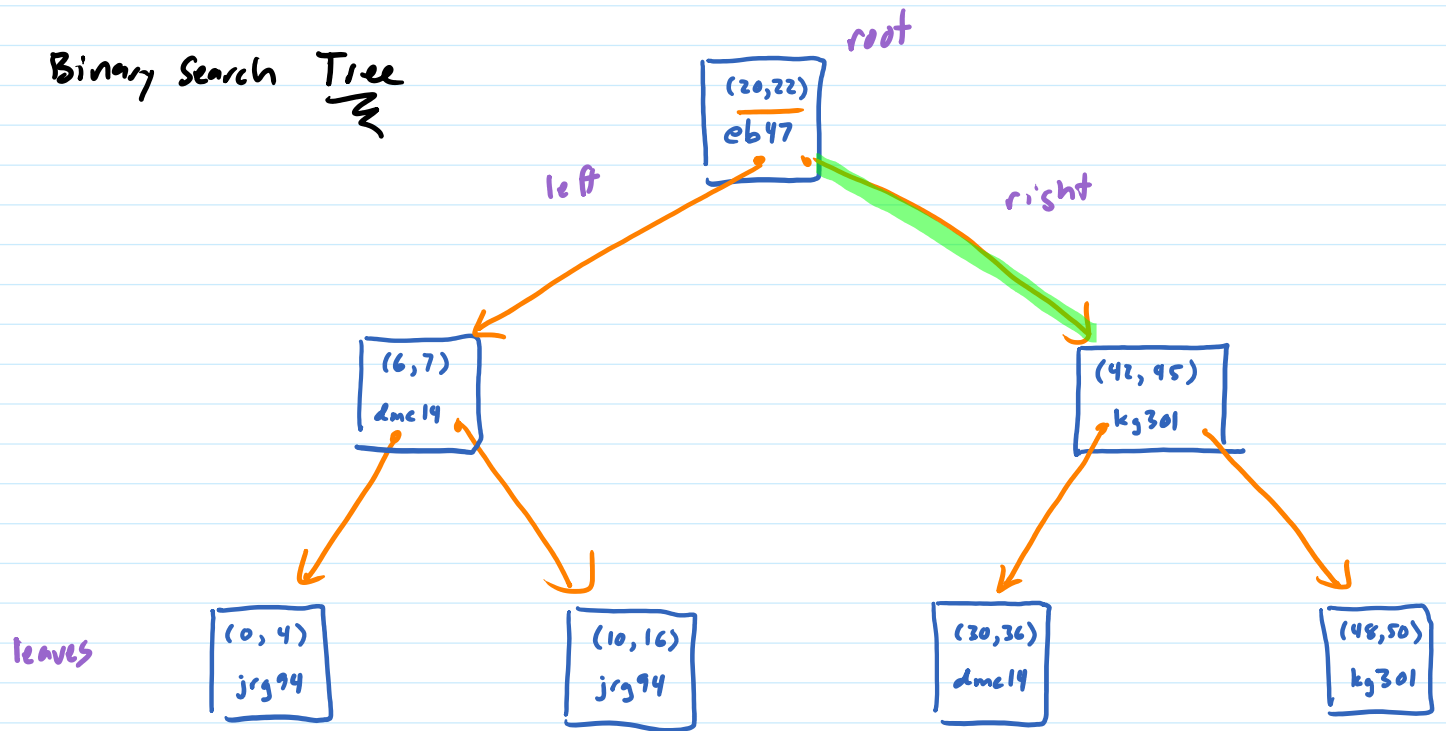


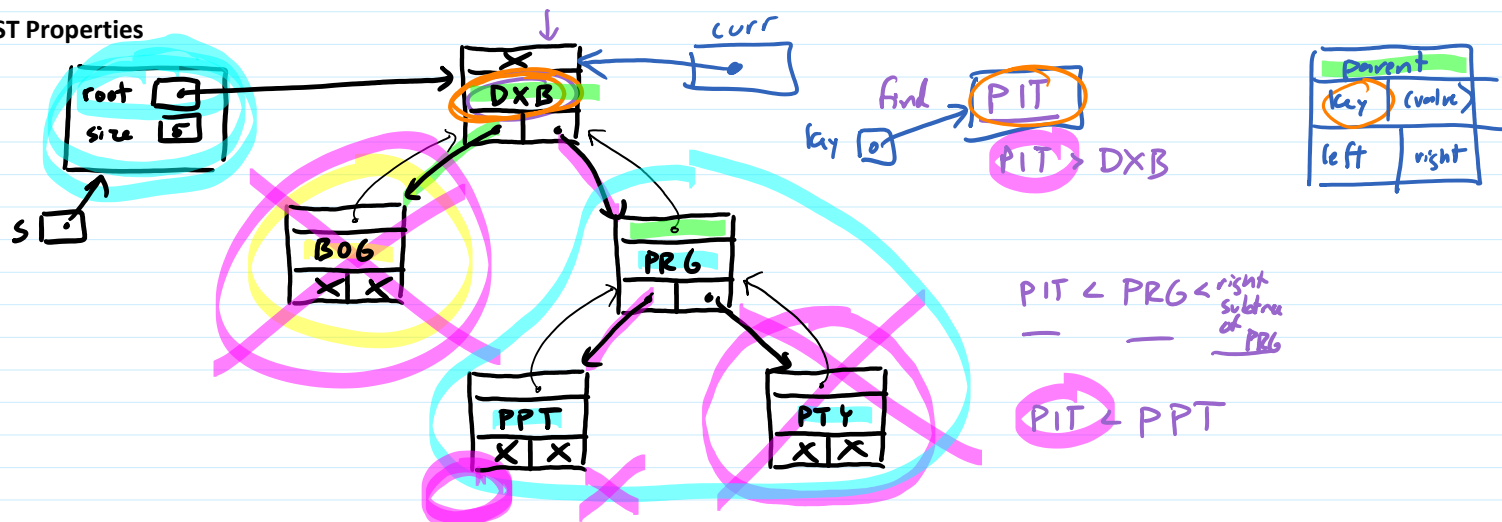
Binary Search Trees



Binary Search Tree



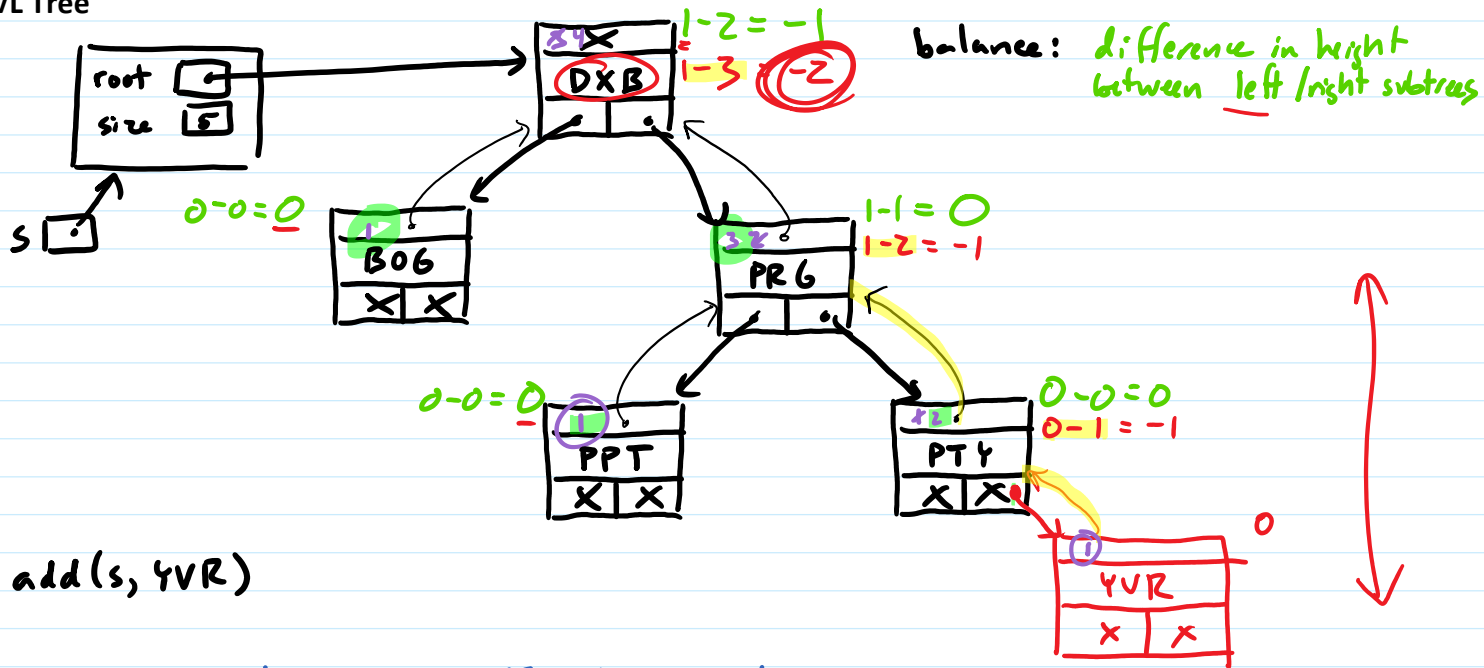
BST Properties



Properties: for every node n , keys in left subtree are $<$ key in n
 keys in right subtree are $>$ key in n

```
bool smap_contains_key(smap *m, const char *key)
{
    smap_node *curr = m->root;
    while (curr != NULL && strcmp(key, curr->key) != 0)
    {
        if (strcmp(key, curr->key) < 0)
        {
            curr = curr->left;
        }
        else
        {
            curr = curr->right;
        }
    }
    return (curr != NULL);
}
```

AVL Tree



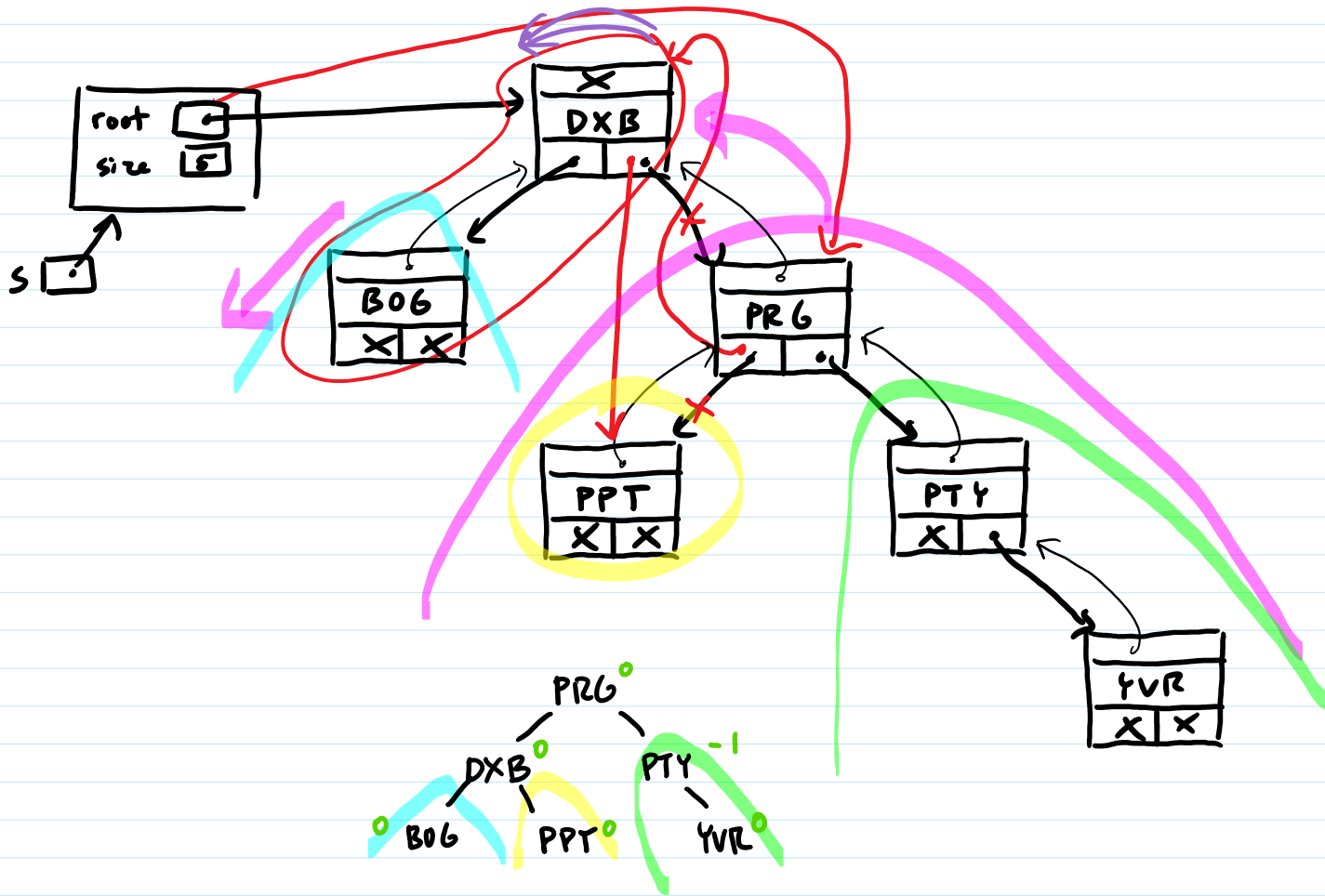
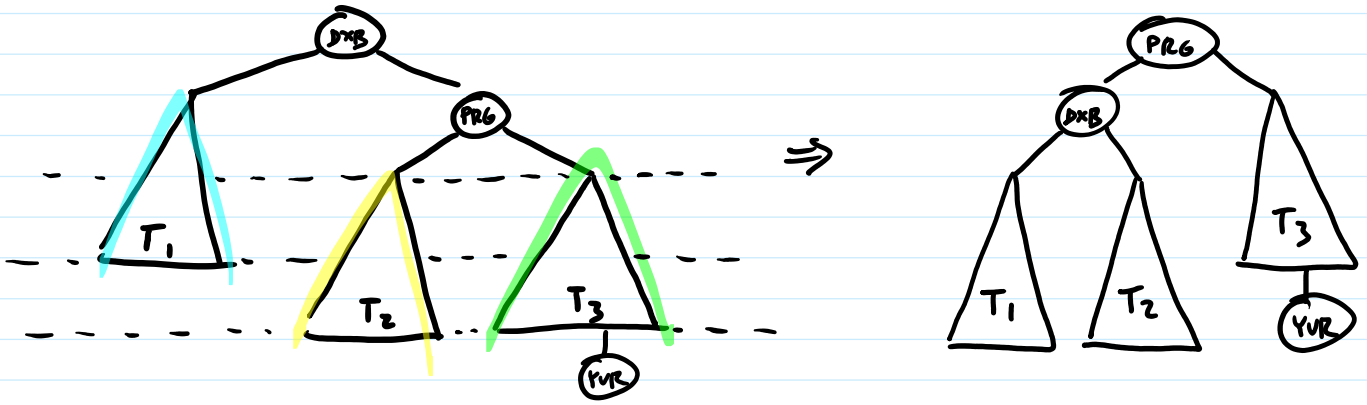
add(s, YVR)

properties: 1) same BST order property

2) all nodes have balance -1, 0, or 1

↳ guarantees that h is $O(\log n)$

Rotations



Single/Double Rotation

