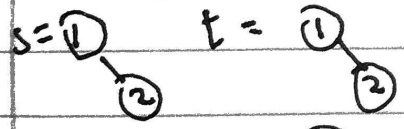
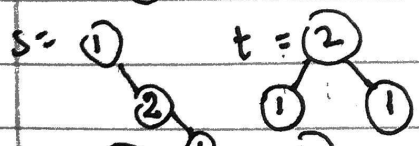
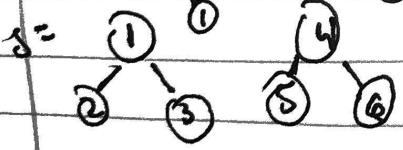
WEEK 6

TALK

- input? 2 non-empty binary trees s and t, root nodes
- return type? boolean
- are duplicates allowed? yes
- t is null? return true
- t and s can be the same
- s is null? true if t is null, else false
- node has left and right node - Binary Tree

EXAMPLES

IN	CLASS	OUT
s=null, t= ①	s is null	false
s = ①, t = null	t is null	true
s= null, t = null	s and t are null	true
	same tree	true
	same values, different structure	false
	same structure, different values	false

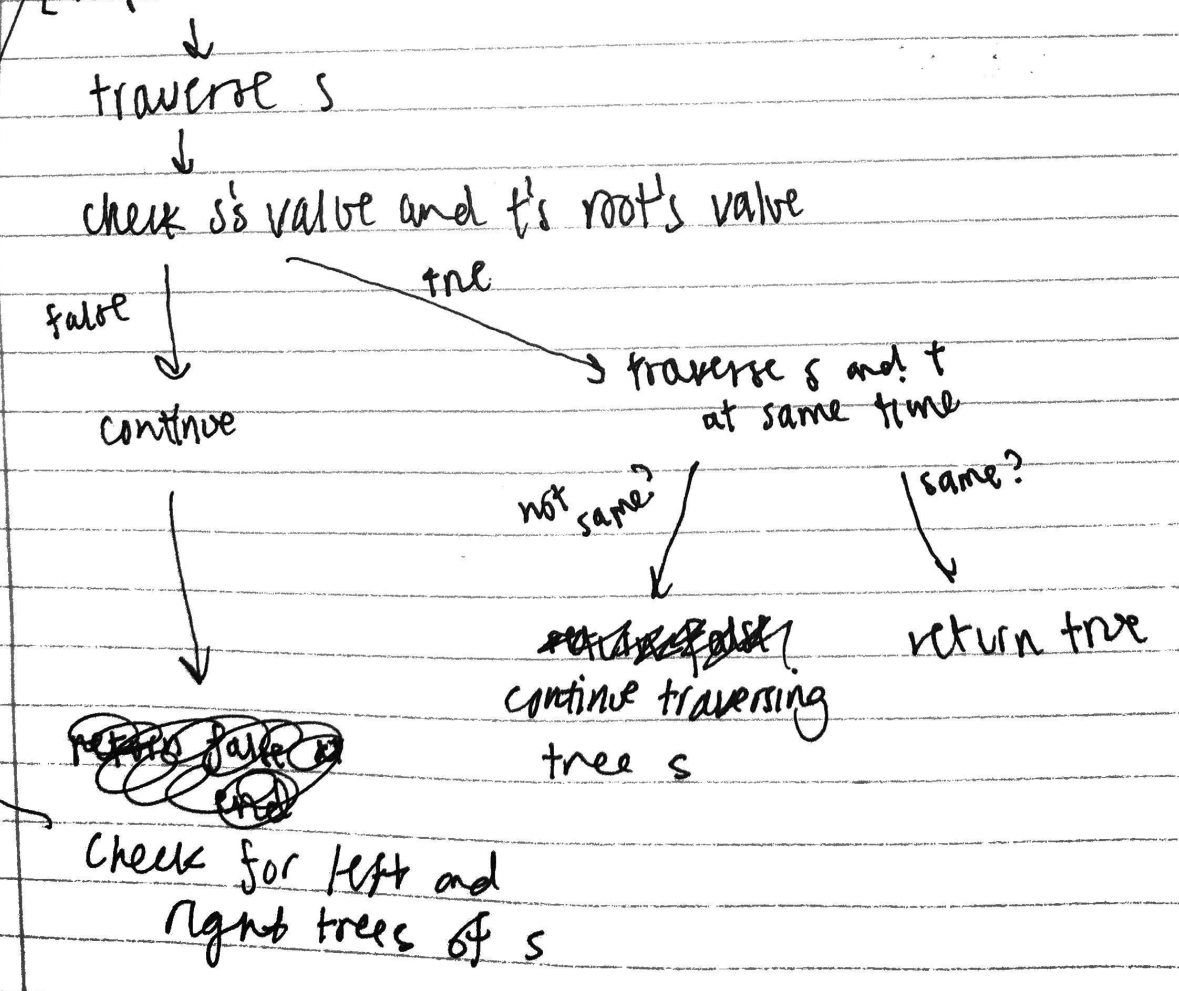
BRUTE FORCE

traverse s and copy to array in preorder
traverse t and copy to array in preorder
check if t is a subarray of s

OPTIMIZE

traverse s in preorder
if a node in s has same value as t's root node,
traverse t at same time and check if same tree

WALK THROUGH



Implementation

```
isSubtree(Node s, Node t):
    if s == null:
        return false
    if t == null
        return true

    if isSame(s, t)
        return true
    else:
        return isSubtree(s.left, t) || isSubtree(s.right, t)

isSame(s, t)
    if s == null && t == null
        return true
    if s == null || t == null
        return false
    return s.val == t.val && isSame(s.left, t.left)
                          && isSame(s.right, t.right)
```

TEST

IN	CLASS	OUT
s = null, t = ①	s is null	false
s = ①, t = null	t is null	true
s = null, t = null	both s and t are null	true
s = ② ①, t = ② ①	same tree	true
s = ④ ②, t = ⑦ ④	same values, different structure	false
s = ⑤ ④ ⑪ ④, t = ⑧ ⑩ ⑨	same structure, different values	false