

Hash table insert vs remove.

Insert (1, 2)

(1, 3)

Worst-case  
 $O(1)$  time



lookup(1) always  
finds this.

remove()

Only remove (1, 3)

or  
all Pairs (1, x)

Shadowing

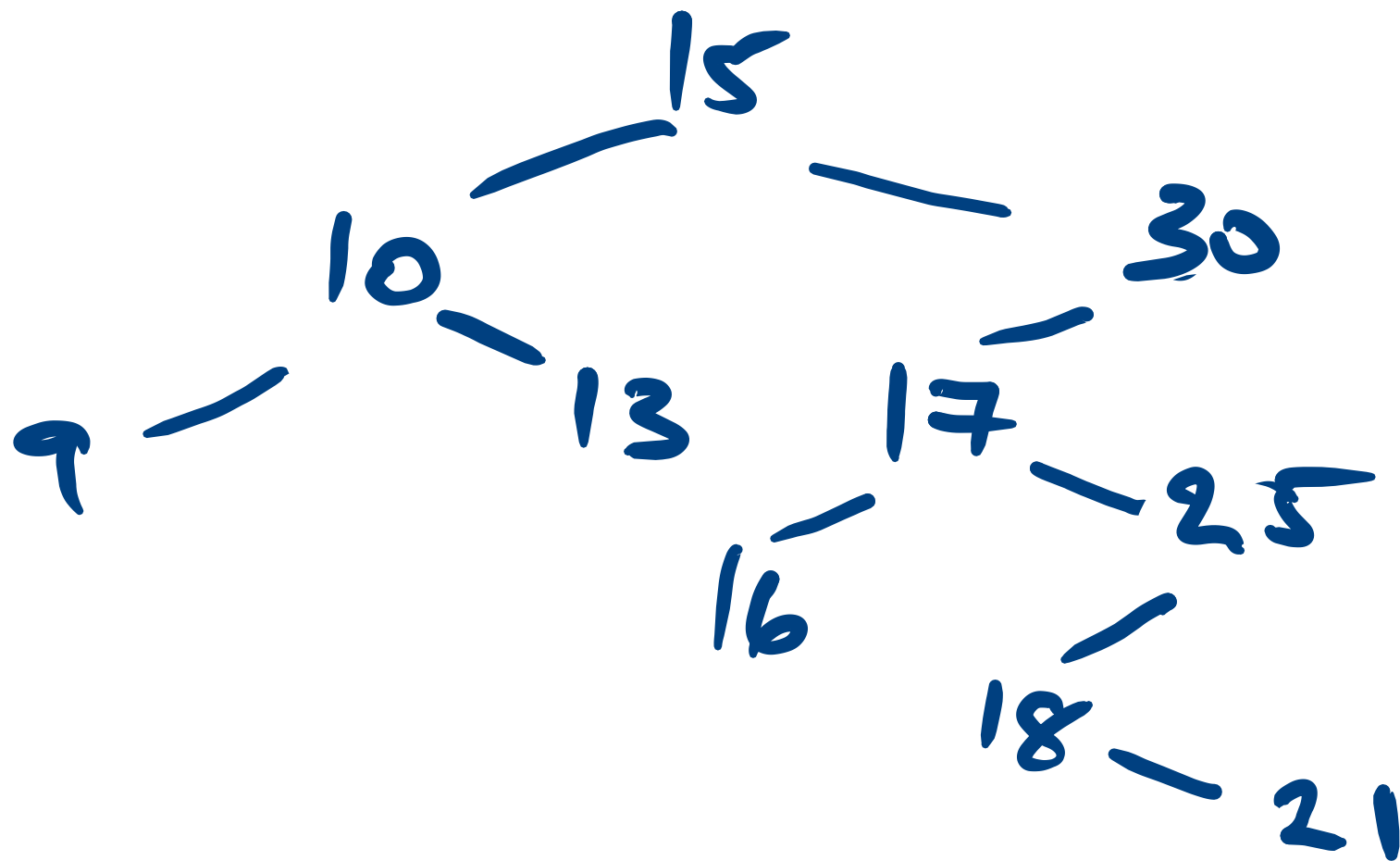
the first pair } expected  
 $O(1)$   
time.

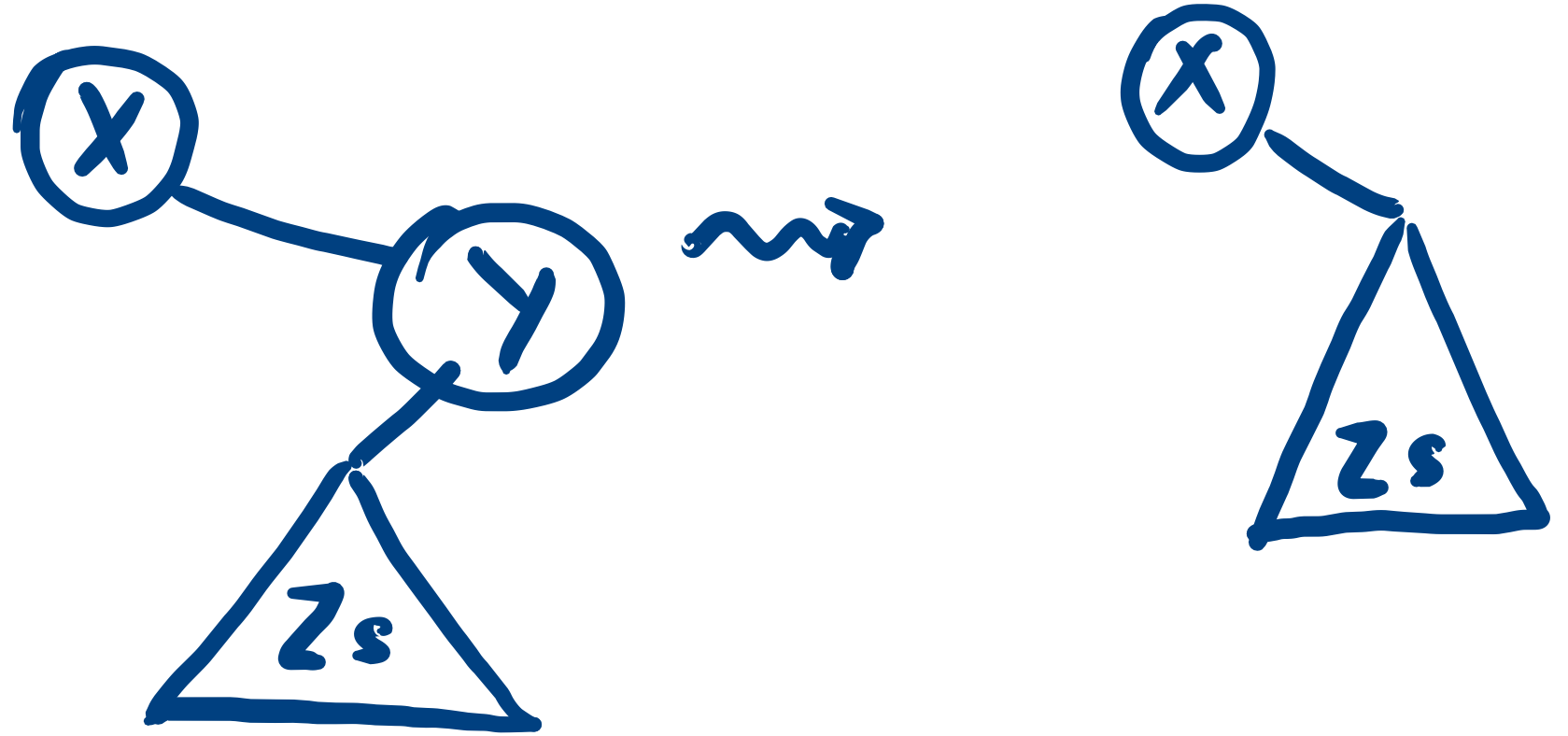
BST

remove()

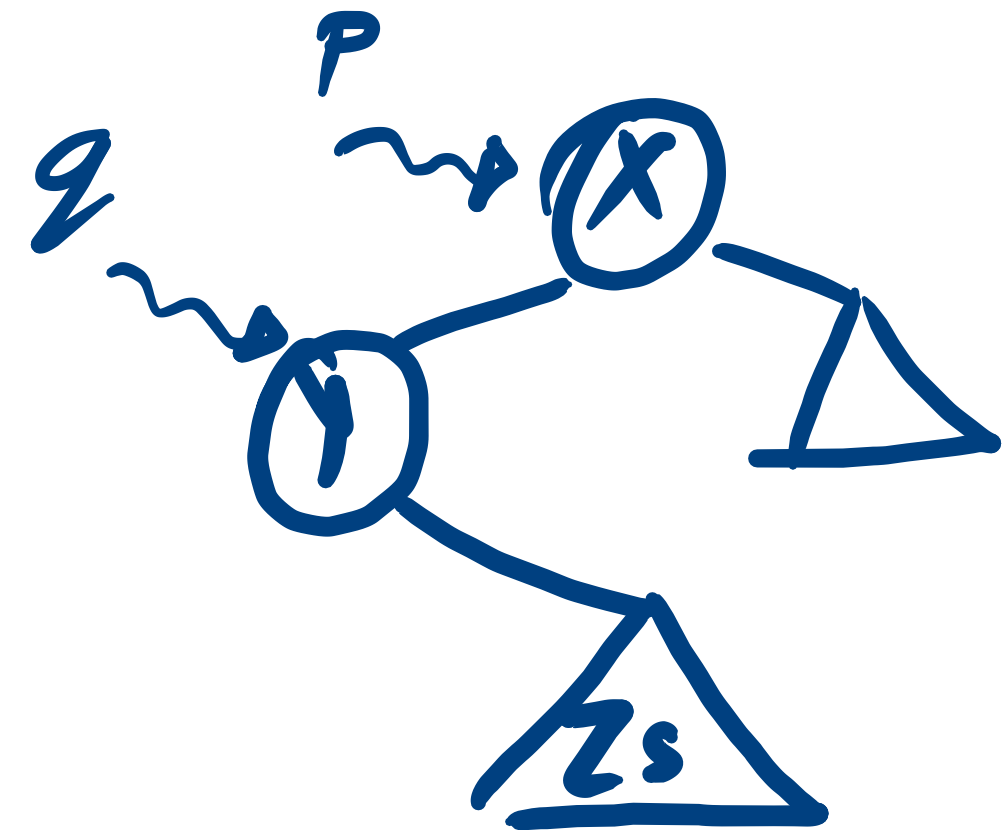
• Key = 13

replace  
10's right  
with NULL

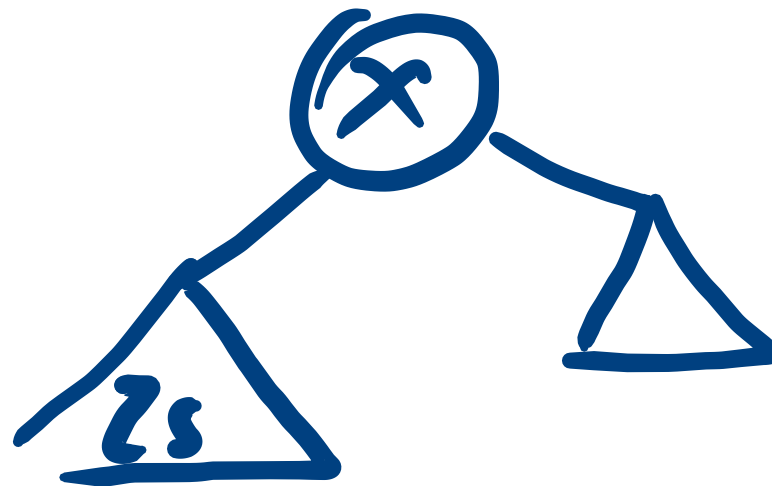




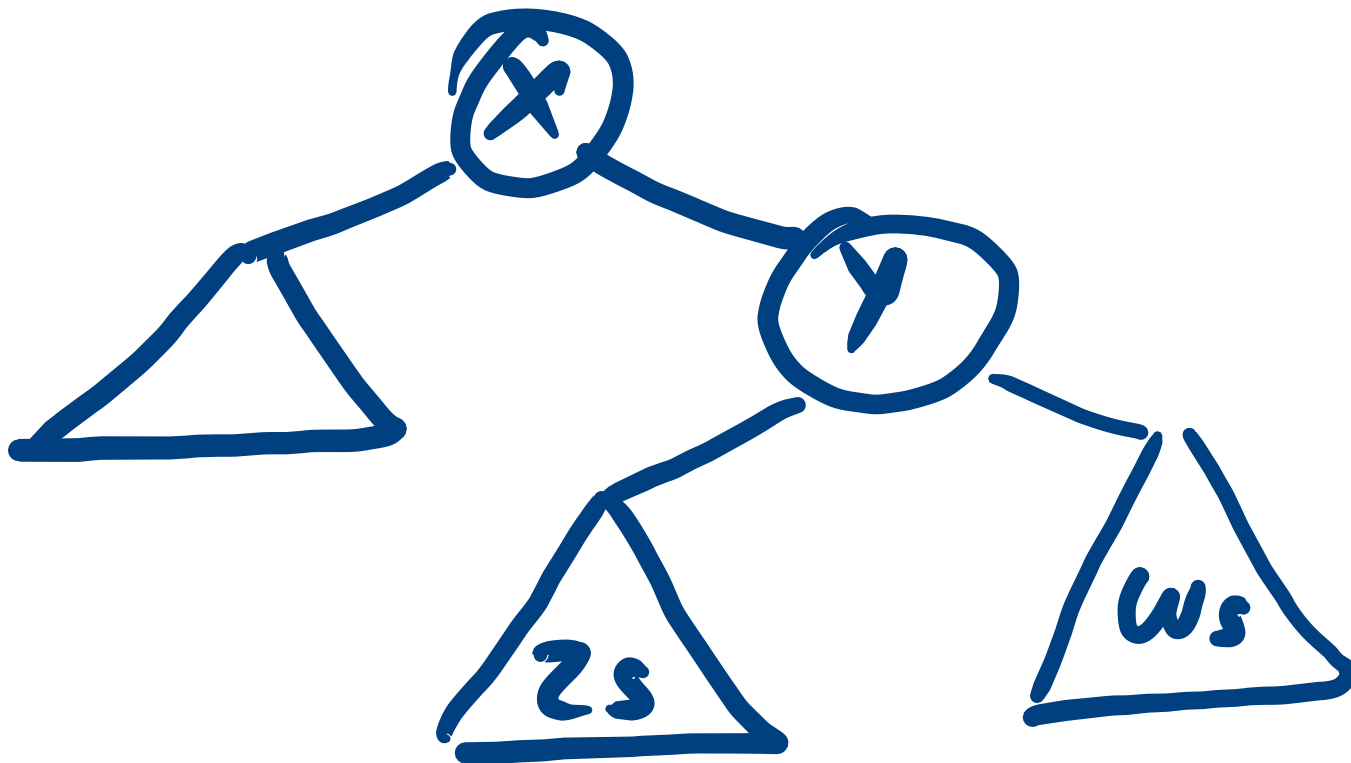
We remove  $y$  by setting  $x \cdot \text{right}$  to point to the root of  $Zs$ .

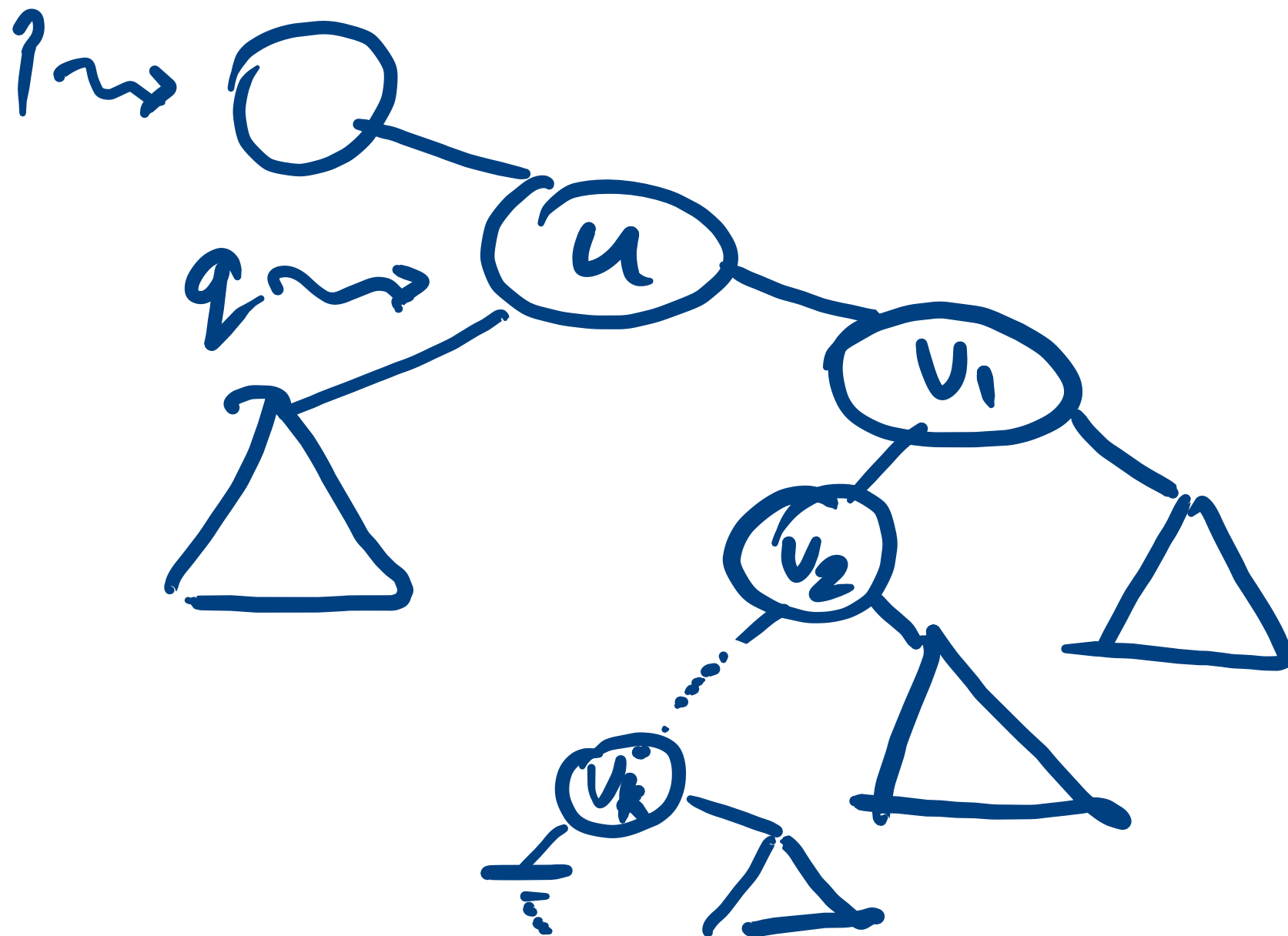


~>



remove(y)

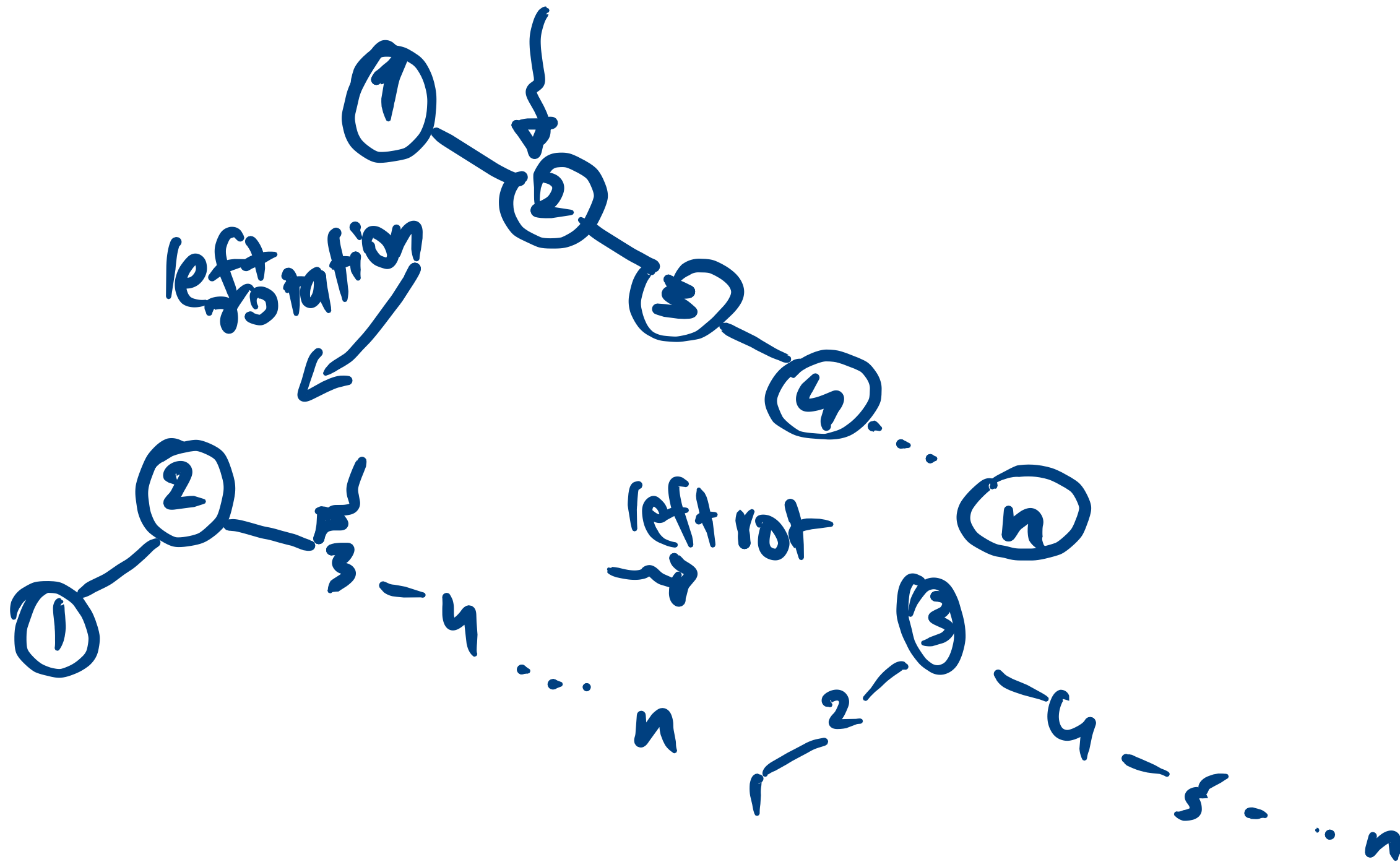




BSTs have bad and good  
insertion sequences

Soln.

- Randomized insertion sequence
- Balanced BST
  - AVL trees
  - Red-Black trees.







# Summary (DSA1)

Model problems using graphs.

{ DFS } whole graph exploration.  
{ BFS }

In DSA2 ~ Design of Problem-Specific algorithms

In DSA1 ADT

Stacks

Queues

dictionaries

hash tables

BSTs

Implementation

Arrays  
Ring buffers  
linked lists:

pointer based  
data structures.

suffix trees  
tries

Databases

B-tree

Red-Black trees

Fibonacci heaps, Binary heaps,  
t-reaps,