

Rubin Peci
 CS 201 - Data Structures
 December 6th, 2019

A.1

R-9.10

In a min heap, you can find the third smallest key at the first level, or the children of the parent node, since that subtree will have, at minimum, 3 keys, and one of them will be the third lowest.

R-9.11

In a min heap, the largest key would be stored on the very bottom level of the tree, or as a leaf.

A.4

```
int ht(Bnode2 u) {
    if (u == null) {
        return -1;
    } else {
        int leftHeight = ht(u.left);
        int rightHeight = ht(u.right);

        if (leftHeight > rightHeight)
            return leftHeight + 1;
        else
            return rightHeight + 1;
    }
}
```

A.5

v=u.parent should be updated by the following conditions
 if the new node added ends up in the left subtree,
 then the balance will increase by 1
 if the new node added ends up in the right subtree
 then the balance will decrease by 1

A.7

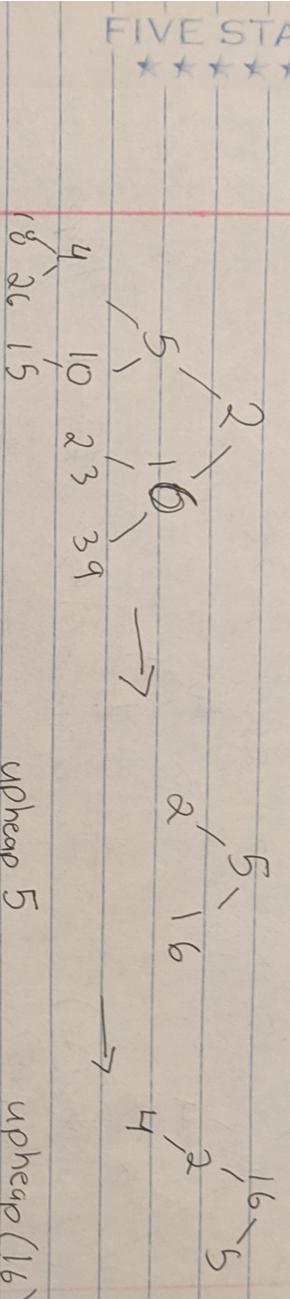
```
Node split(Node u) {
    if u is null or u.next is null
        return u
    else
        tmp node = u
        length = 0
        while tmp != null
            tmp = tmp.next
            length++
        newTmp node = u
        for ( i < length / 2)
            newTmp = newTmp.next

        tmp = u
        while tmp != null
            if tmp.next = newTmp
                tmp.next = null
                break loop
            tmp = tmp.next

        return newTmp
}
```

A.2

(2, 5, 16, 4, 10, 23, 39, 18, 26, 15)

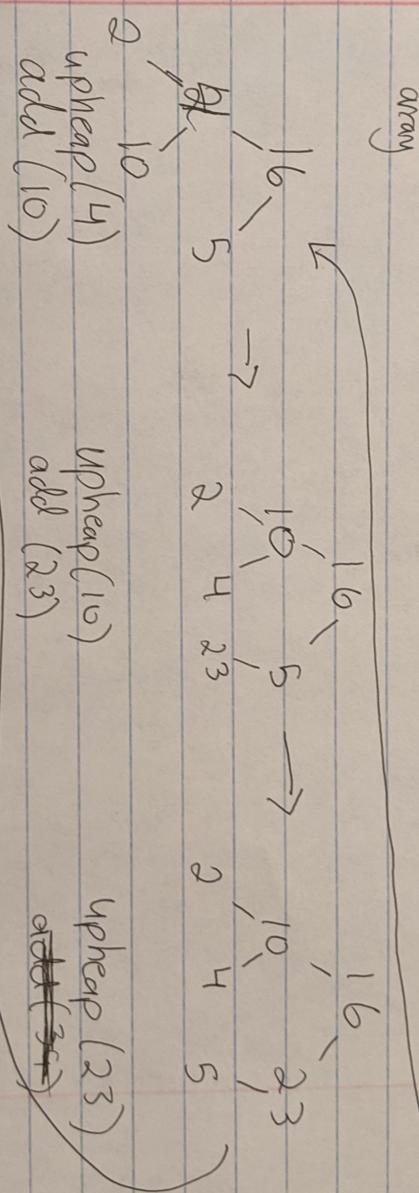


Initially adding the

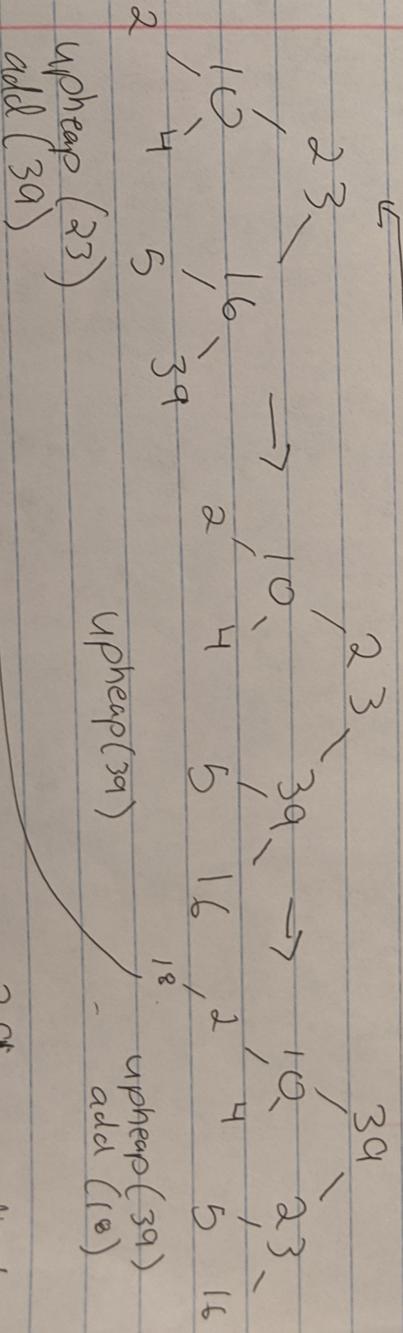
array

upheap(5)
add(16)

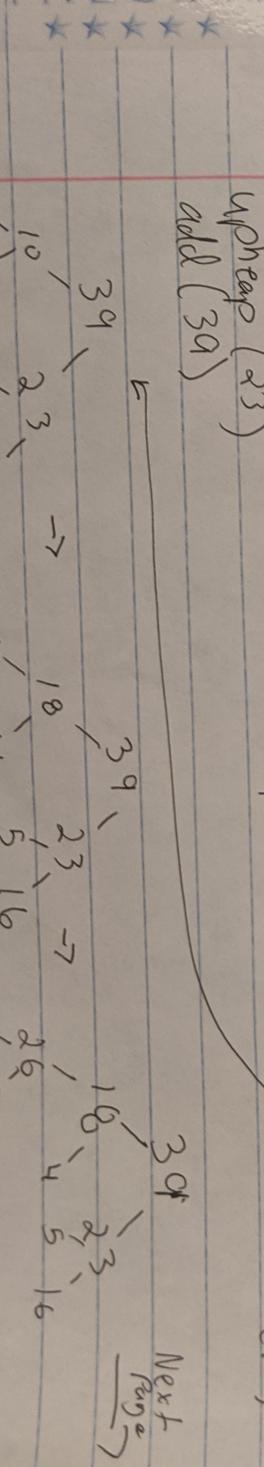
upheap(16)
add(4)

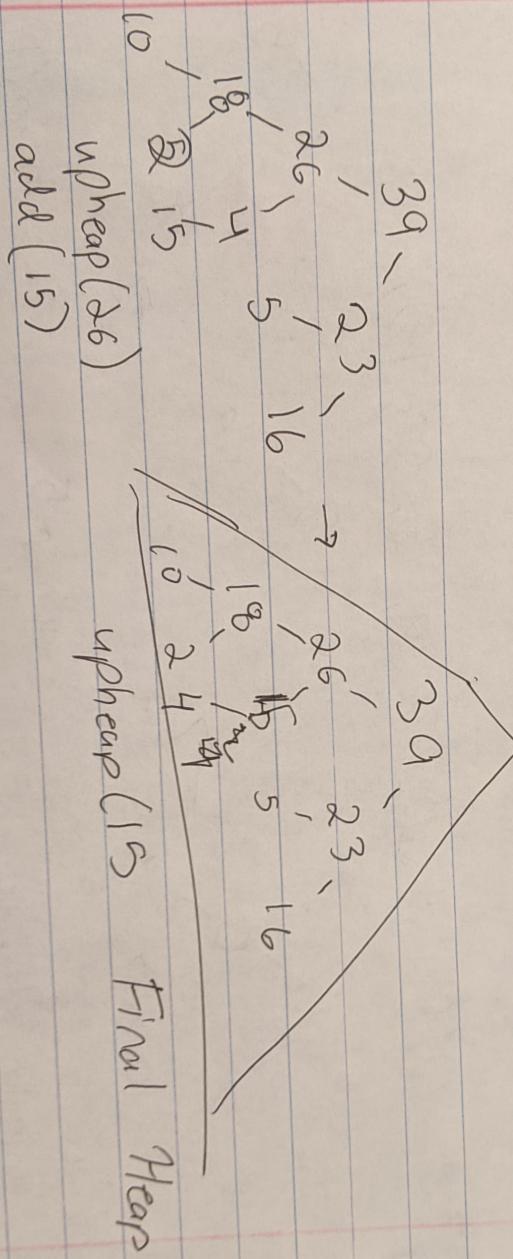


FIVE STAR.

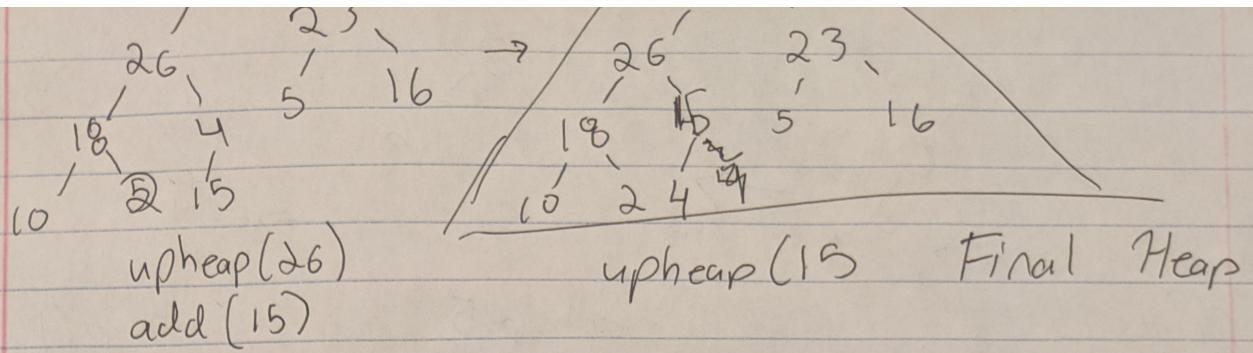


FIVE STAR.

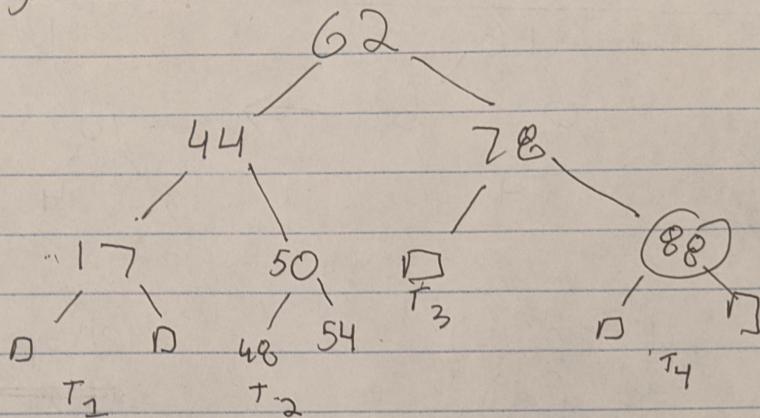




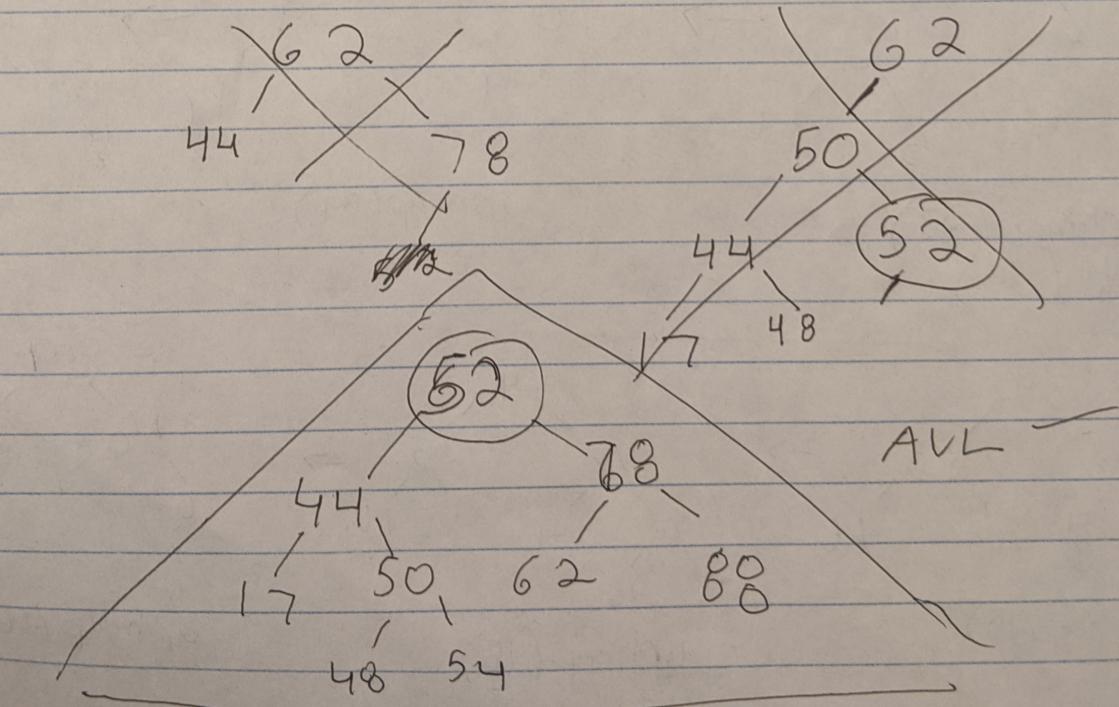
upheap(26) upheap(15) Final Heap
add(15)



A.3 Original



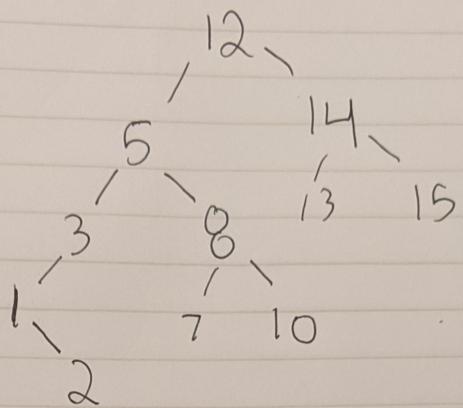
Adding 52 ... to ~~T₃~~! to ~~T₁ + T₂~~ $T_3 + T_4$



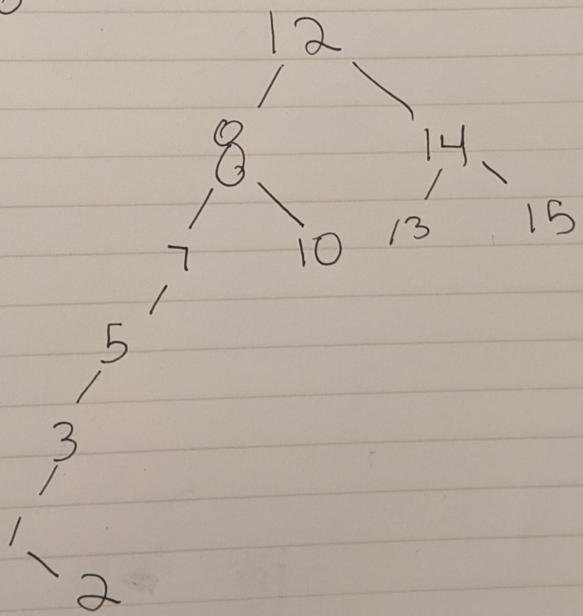
A. 6

2, 1, 3, 7, 10, 8, 5, 13, 15, 14, 12

a)



b) Rotate 8



c) Post-order traversal

2, 1, 3, 5, 7, 10, 8, 13, 15, 12