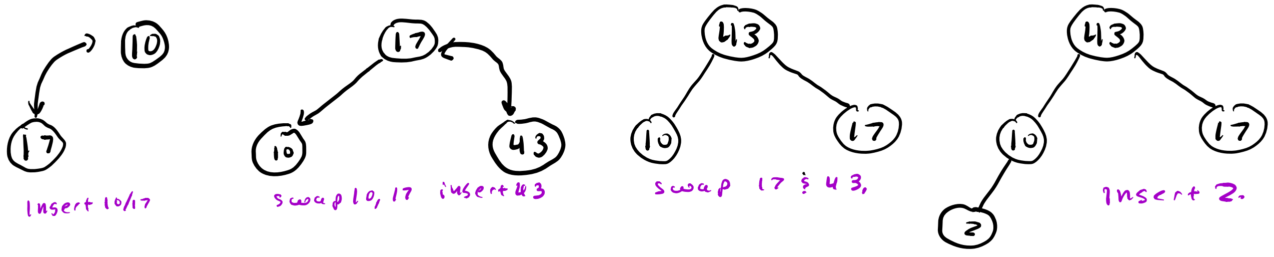
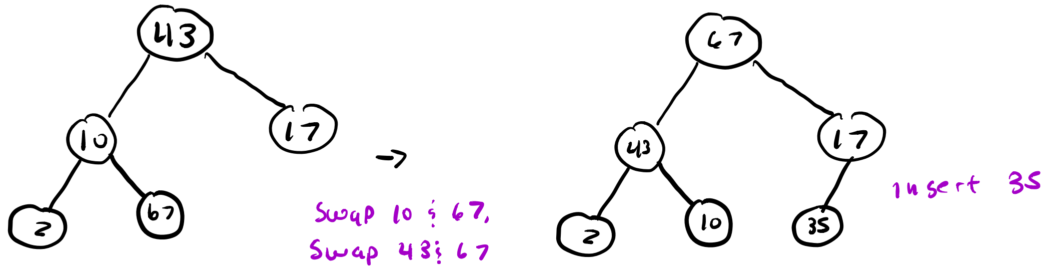
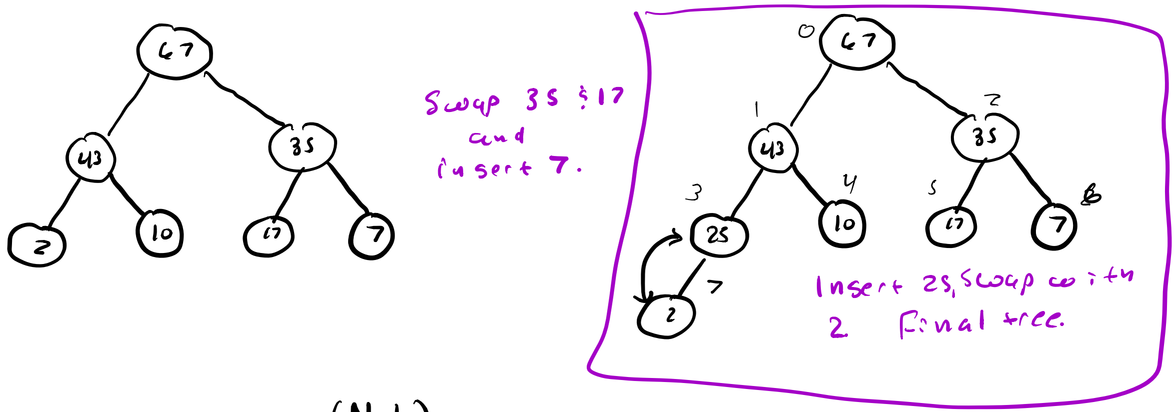
1) a) 10, 17, 43, 2, 67, 35, 7, 25







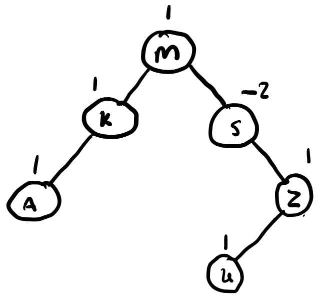
b) Parents index = $\left(\frac{N-1}{2}\right)$, parent of node 25: $\left(\frac{3-1}{2}\right) = 1 = 43$ ✓

c) Heapsort on a max heap to order in non-decreasing order:

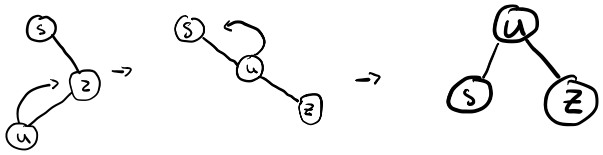
- First you need to create the heap for a given array which forms a valid heap
- Then you need to apply root deletion n-1 times on the remaining heap. The first node you delete is the last in the array, last to delete is first.

2) a) A BST needs to be balanced at all times because the difference in the heights of the nodes of left and right subtrees can never be greater than 1. If it is higher than this we perform a rotation to make it balanced.

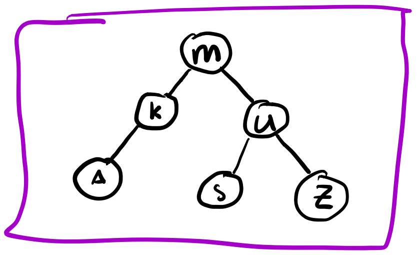
b)



Here, u is unbalanced so we need to do a Right left rotation, starting with S.



Final tree:



3) b)

10, 17, 43, 2, 67, 35, 7, 25

pivot is 2 (smallest values) 2 to end.

i j

10, 17, 43, 25, 67, 35, 7, 2

i stays, j crosses.

Switch 2 & 10

pivot

2, 17, 43, 25, 67, 35, 7, 10

Switch 2 & 10, pivot is now 67.

i j

2, 17, 43, 25, 10, 35, 7, 67

j stays, i crosses.

67 stays, no swaps

i . . i j . . j

2, 17, 43, 7, 10, 35, 25, 67

pivot is 25, move to end

2, 17, 10, 7, 43, 35, 25, 67

43 gets swapped w/ 10.

j < 25, i > 25

2, 17, 10, 7, 25, 35, 43, 67

swap 25 & 43

2, 17, 7, 10, 25, 35, 43, 67

swap 10 & 7

2, 7, 10, 17, 25, 35, 43, 67

swap 7 & 17

swap 17 & 10

more pivots to select but running out of time... 43 then 35, ends up the same

b) when array is already sorted or in reverse order.

Choose a random index (or middle index since random is costly).

4)

1 2 3 4

a) 10, 17, 43, 2, 67, 35, 7, 25

10	17	43	2	67	35	7	25
0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
	3	0	0	0	0	0	0
		6	0	0	0	0	0
			0	0	0	0	0
				7	0	0	0
					5	0	0
						1	0
							4

Array S[0..7]

= 2 7 10 17 25 35 43 67

b)

10	17	43	2	67	35	7	25
1	1	1	1	1	1	1	1
1	2	3	4	5	6	7	8

25

25

Not really, no. Since the values are distinct, they are not distributed across the entire array, Distribution Count works better when the array has similar values.