CSI 2110 Fall 2015
Computer Science University of Ottawa

Assignment #2
Matt Langlois - 7731813
October 25

Question 1

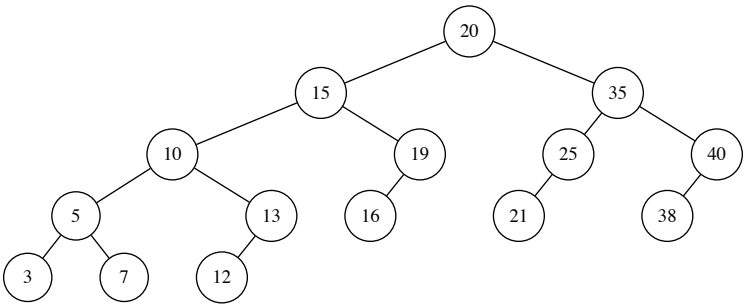
a) 20, 15, 10, 5, 3, 7, 13, 12, 19, 16, 35, 25, 40, 38

b) 3, 7, 5, 12, 13, 10, 16, 19, 15, 25, 38, 40, 35, 20

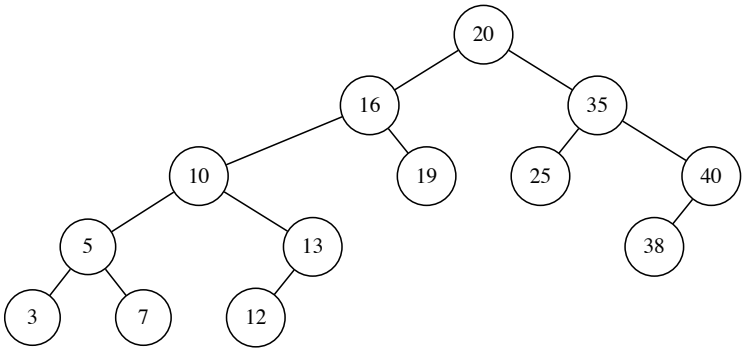
c) Searches for the largest value in a binary tree.

```
findMax(Node n) {
    if (n.hasRightChild()) {
        return findMax(n.rightChild());
    }
    return n;
}
```

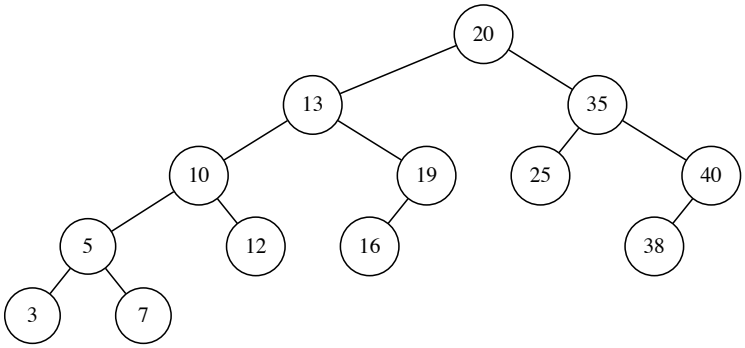
d) *Insert element:* 21



e) Case 1: Replace 15 with the left most node of the right subtree. Then remove the leftmost node of the right subtree.



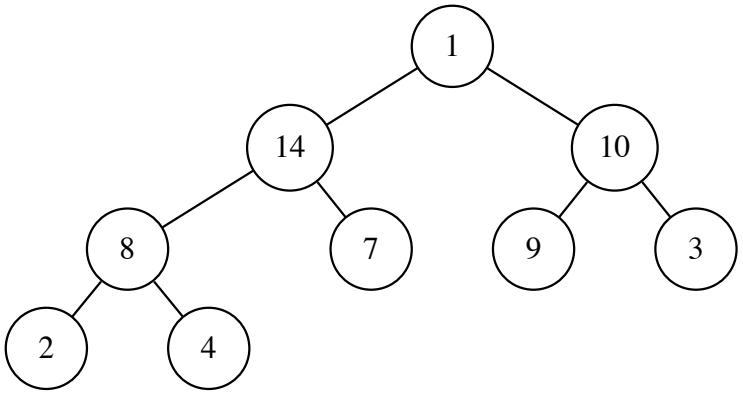
Case 2: Replace 15 with the right most node of the left subtree. Any children of the right most node become children of the right most node's parent.



Question 2

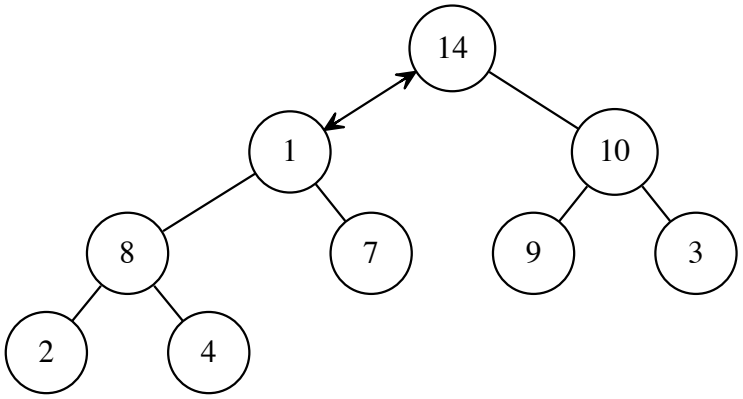
a) Step 1: Replace node with the last node in the heap

1	14	10	8	7	9	3	2	4



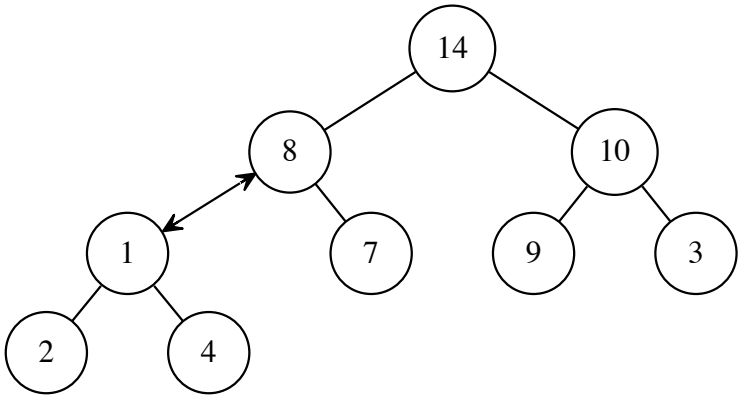
Step 2: Downheap while the children are larger

14	1	10	8	7	9	3	2	4



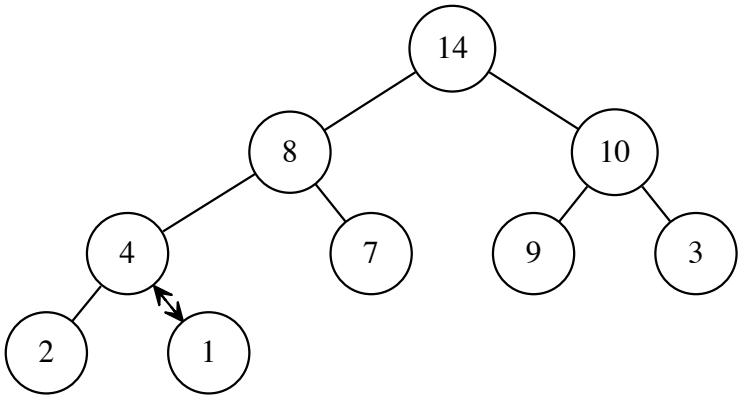
Step 3: Downheap while the children are larger

14	8	10	1	7	9	3	2	4



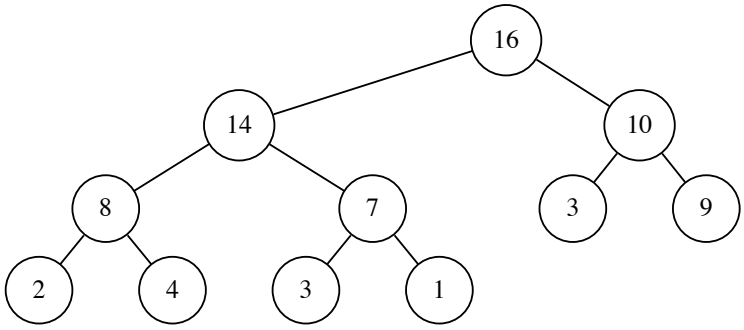
Step 4: Downheap to become a leaf node

14	8	10	4	7	9	3	2	1



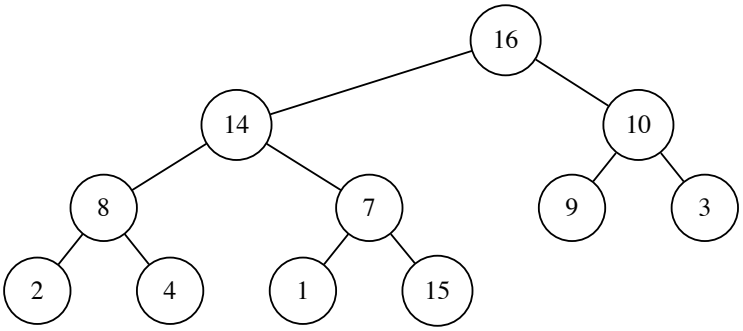
b) Insert element 3 at the left most position on the empty row. No further changes are required as the $7 \geq 3$ property of the max heap is satisfied

16	14	10	8	7	9	3	2	4	1	3



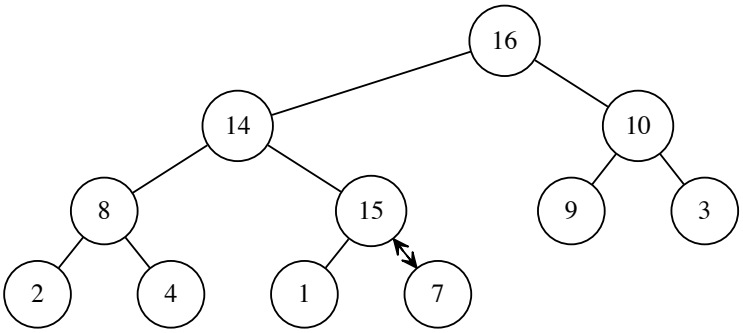
c) Step 1: Insert at left most node in the empty row

16	14	10	8	7	9	3	2	4	1	<u>15</u>



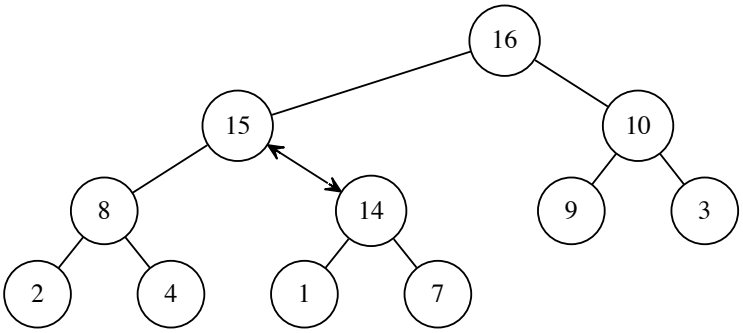
Step 2: Upheap 15 until max heap property *parent* $\geq$ *child* is met

16	14	10	8	<u>15</u>	9	3	2	4	1	<u>7</u>



Step 3: Upheap once more to satisfy the max-heap property.

16	<u>15</u>	10	8	<u>14</u>	9	3	2	4	1	7



d) Calculate the height of a 2000 element heap:

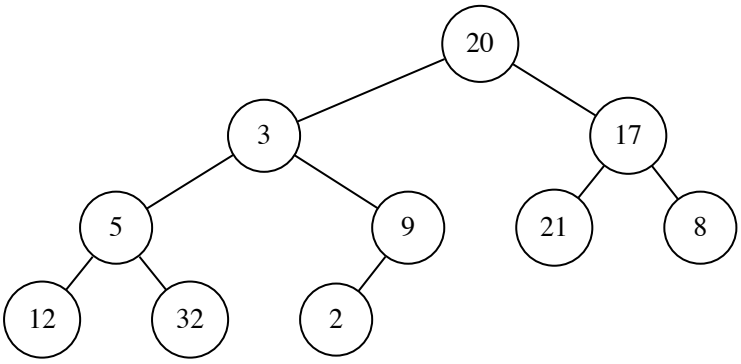
$$h(n) = \lfloor log_2(n) \rfloor$$
$$h(2000) = \lfloor log_2(2000) \rfloor$$
$$h(2000) = 10$$

$\therefore$ the height of a 2000 node heap is 10.

Question 3

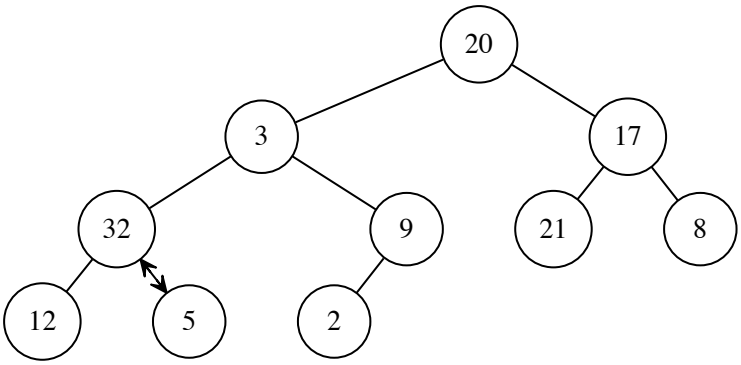
Step 1: Heapify the contents of the array into a max-heap

Heap:	20	3	17	5	9	21	8	12	32	2



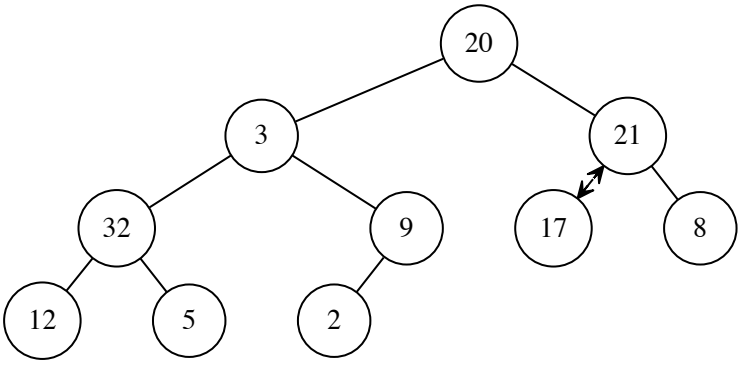
Step 2: Upheap while the parents are smaller

Heap:	20	3	17	<u>32</u>	9	21	8	12	<u>5</u>	2



Step 3: Upheap while the parents are smaller

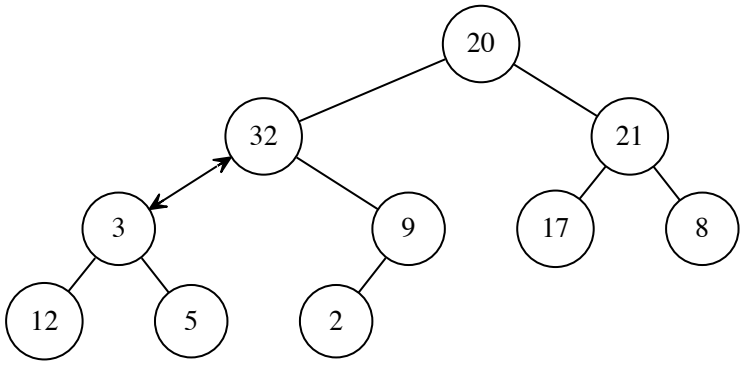
Heap:	20	3	<u>21</u>	32	9	<u>17</u>	8	12	5	2



Step 4: Upheap while the parents are smaller

Heap:

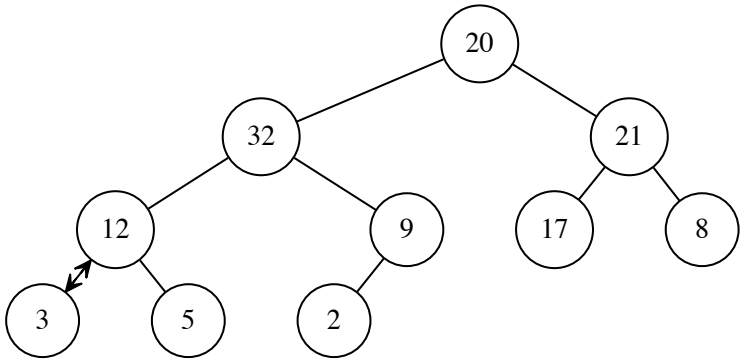
20	32	21	3	9	17	8	12	5	2



Step 5: Upheap while the parents are smaller

Heap:

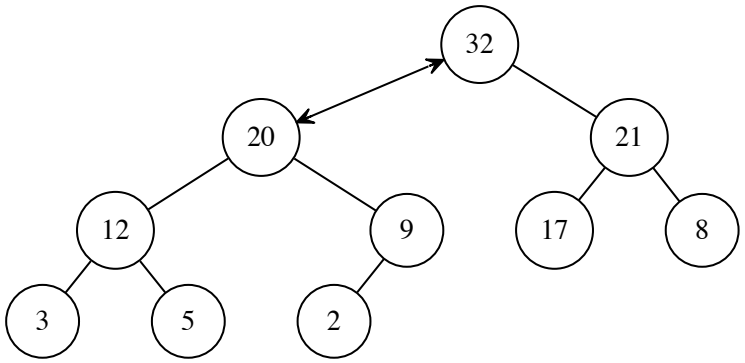
20	32	21	12	9	17	8	3	5	2



Step 6: Form a max heap

Max-heap:

32	20	21	12	9	17	8	3	5	2

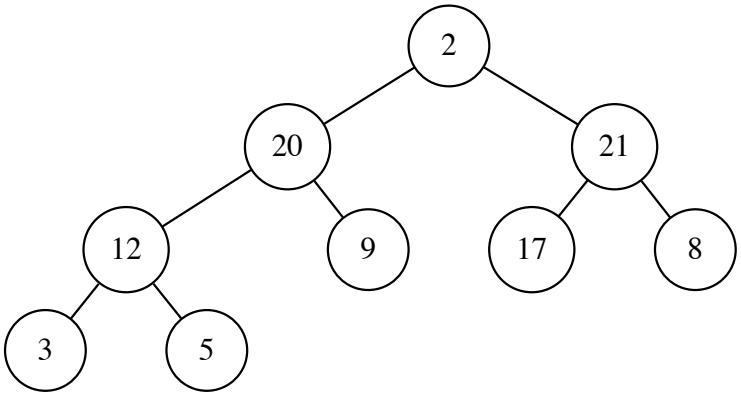


Step 7: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

2	20	21	12	9	17	8	3	5

Sorted Sequence: 32

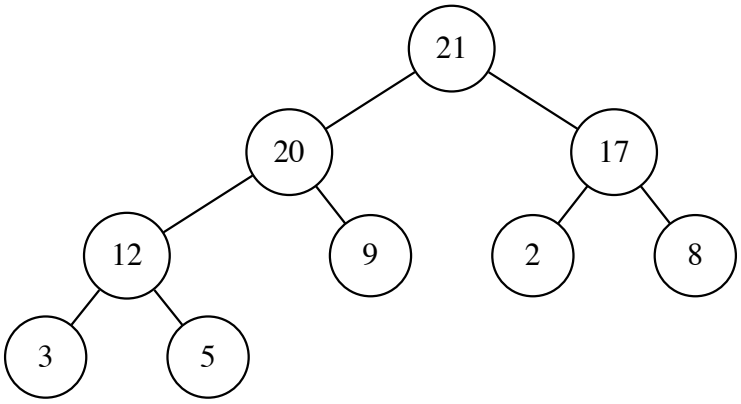


Step 8: Form a max-heap with the remaining elements

Max-heap:

21	20	17	12	9	2	8	3	5

Sorted Sequence: 32



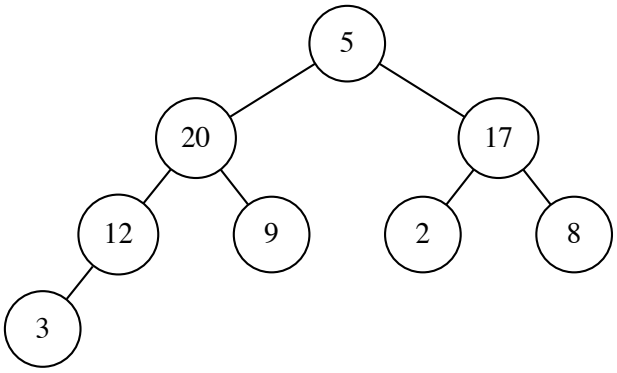
Step 9: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

5	20	17	12	9	2	8	3

Sorted Sequence:

21	32



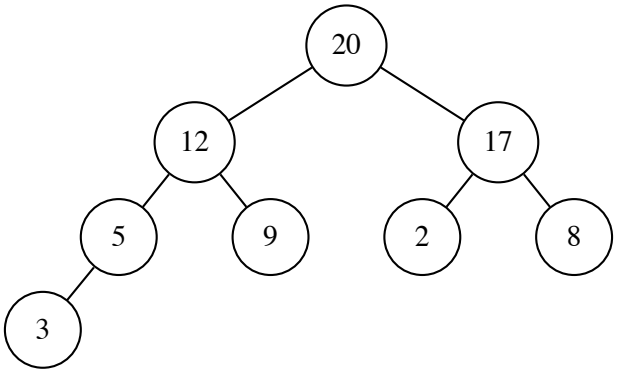
Step 10: Form a max-heap with the remaining elements

Max-heap:

20	12	17	5	9	2	8	3

Sorted Sequence:

21	32



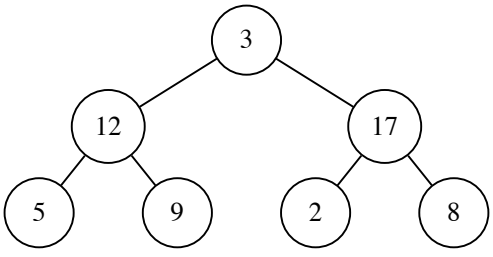
Step 11: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

3	12	17	5	9	2	8

Sorted Sequence:

20	21	32



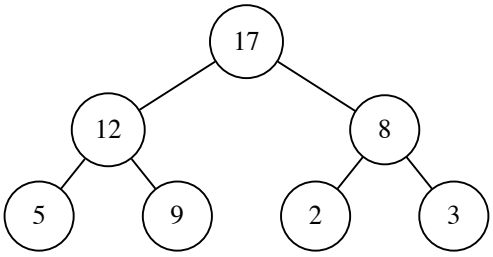
Step 12: Form a max-heap with the remaining elements

Max-heap:

17	12	8	5	9	2	3

Sorted Sequence:

20	21	32



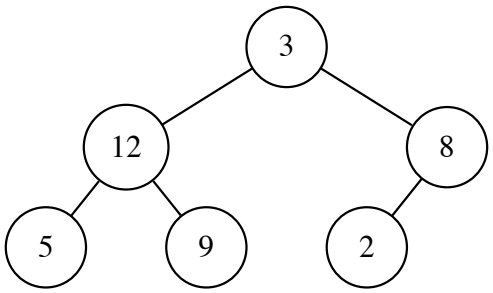
Step 13: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

3	12	8	5	9	2

Sorted Sequence:

17	20	21	32



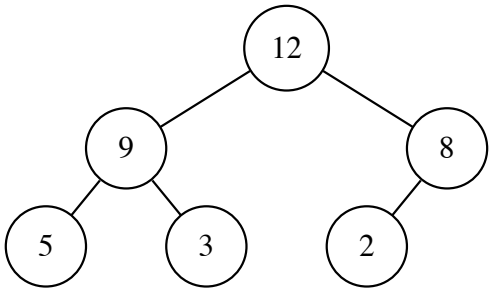
Step 14: Form a max-heap with the remaining elements

Max-heap:

12	9	8	5	3	2

Sorted Sequence:

17	20	21	32



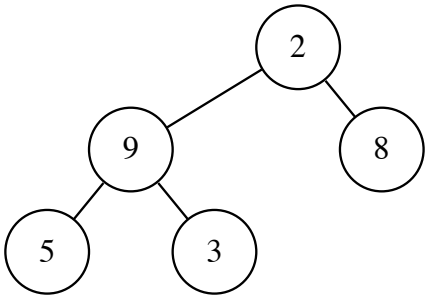
Step 15: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

2	9	8	5	3

Sorted Sequence:

12	17	20	21	32



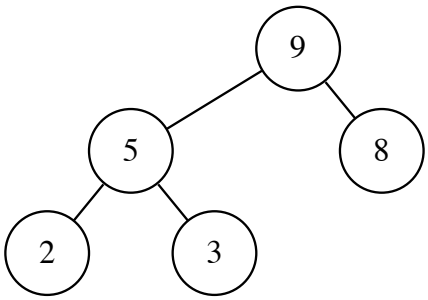
Step 16: Form a max-heap with the remaining elements

Max-heap:

9	5	8	2	3

Sorted Sequence:

12	17	20	21	32



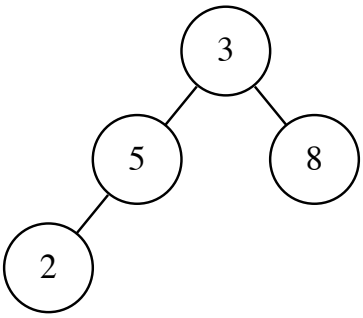
Step 17: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

3	5	8	2

Sorted Sequence:

9	12	17	20	21	32



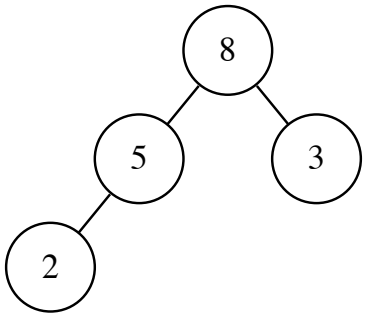
Step 18: Form a max-heap with the remaining elements

Max-heap:

8	5	3	2

Sorted Sequence:

9	12	17	20	21	32



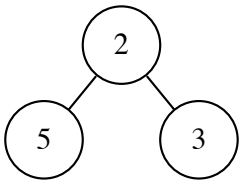
Step 19: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

2	5	3

Sorted Sequence:

8	9	12	17	20	21	32



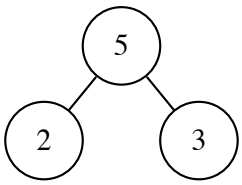
Step 20: Form a max-heap with the remaining elements

Max-heap:

5	2	3

Sorted Sequence:

8	9	12	17	20	21	32



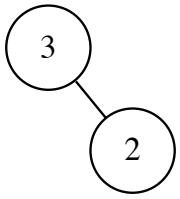
Step 21: Remove the max from the heap and add it to the front of the sorted sequence

Heap:

3	2

Sorted Sequence:

5	8	9	12	17	20	21	32



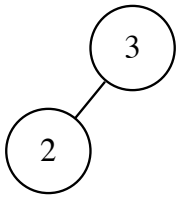
Step 22: Form a max-heap with the remaining elements

Max-heap:

3	2

Sorted Sequence:

5	8	9	12	17	20	21	32



Step 23: Remove the max from the heap and add it to the front of the sorted sequence

Max-heap:

2

Sorted Sequence:

3	5	8	9	12	17	20	21	32

Step 24: Take the final element and add it to the sorted sequence to complete the heapsort

Sorted Sequence:	2	3	5	8	9	12	17	20	21	32

Sorted sequence after heapsort is complete: