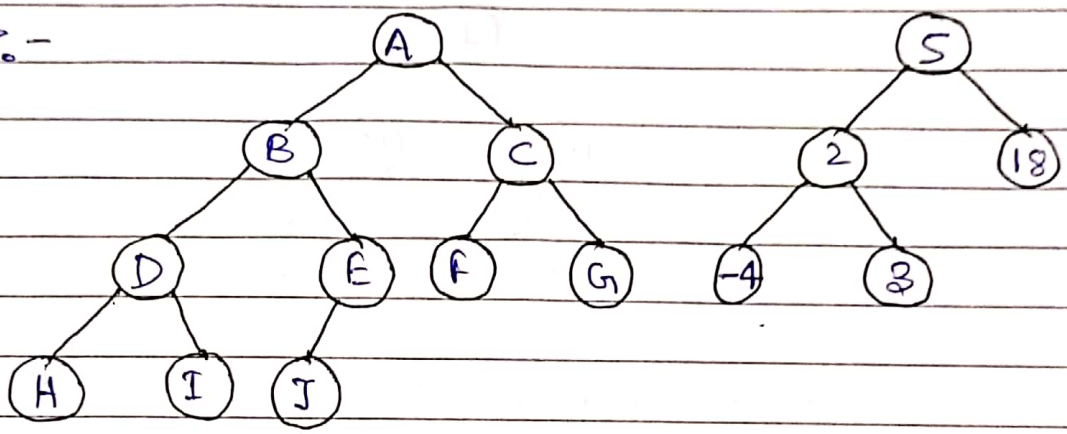
Name - Ravi Prokash singh
sap - 500076081
Roll - R177219145 Batch-5 __/__/____

(1) How is a complete binary tree and perfect different from each other? give example.

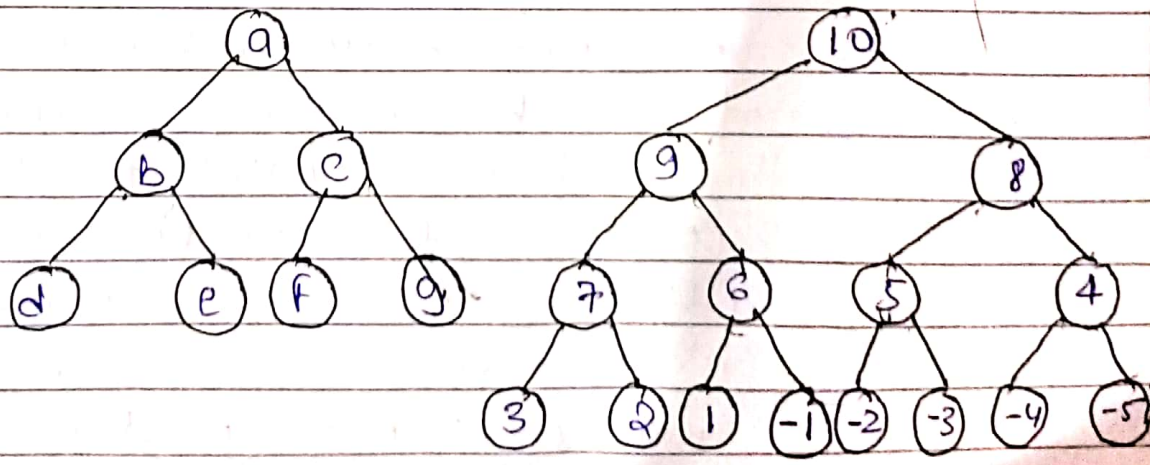
Ans:- complete binary tree is a binary tree in which every level (except possibly the last) is completely filled, and all nodes are as far left as possible

eg:-



Whereas,
A perfect binary tree is a binary tree in which all leaves have the same depth or same level. in other words, Every node has exactly two nodes and all levels are completely filled.

eg:-

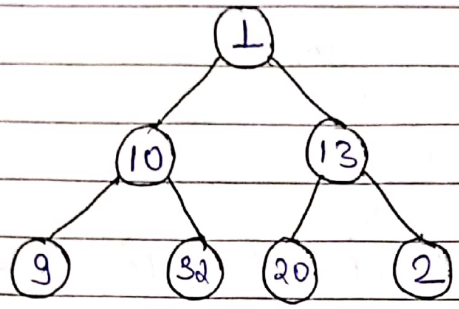


Ravi Prakash Singh
R177219145

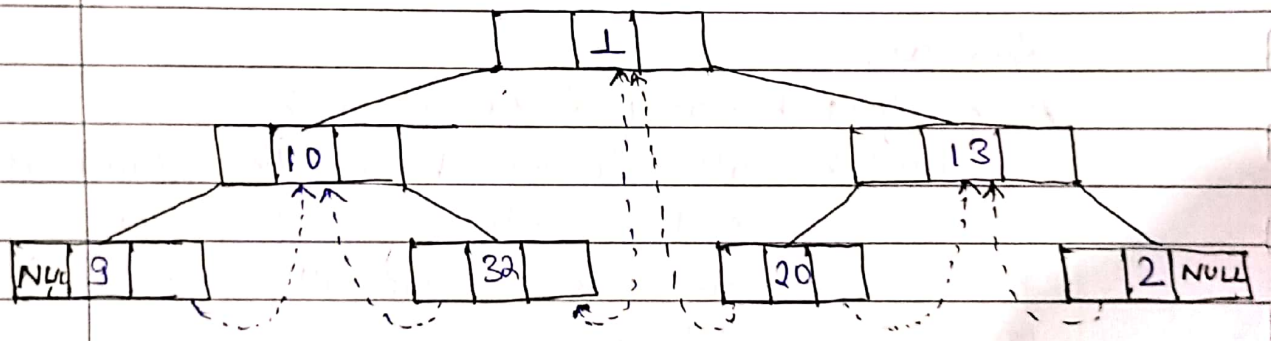
__/__/____

2. Construct a binary tree with the breadth-first traversal order 1, 10, 13, 9, 32, 20, 2. Create a threaded binary tree and state how the threaded binary tree is better than a Binary tree.

Ans:- Binary tree with BFT order - 1, 10, 13, 9, 32, 20, 2



Threaded binary tree



The idea of threaded binary is to make the traversal easy and without making the use of stack or any recursion.
In this, all right NULL pointers are made to point to its inorder successor and left NULL pointer are made to poit to its inorder predecessor. (except the left and right most pointers which should always be NULL). The threaded are also usefull in fast accessing ancestors

Ravi Prakash singh
R177219145

In binary tree all the NULL pointer memory is wasted but in threaded binary tree it make the use of NULL pointer to make the traversal of Next or previous node easier and faster

(3) consider the implementation of hash table using an array of 9 linked list and seperate chaining method of collision avoidance.

(i) Insert the following 9 keys (5, 28, 19, 15, 20, 33, 12, 17, 10) in the given order into the hash table using h(k) = k mod 9 as hash function and draw the final implementation neatly.

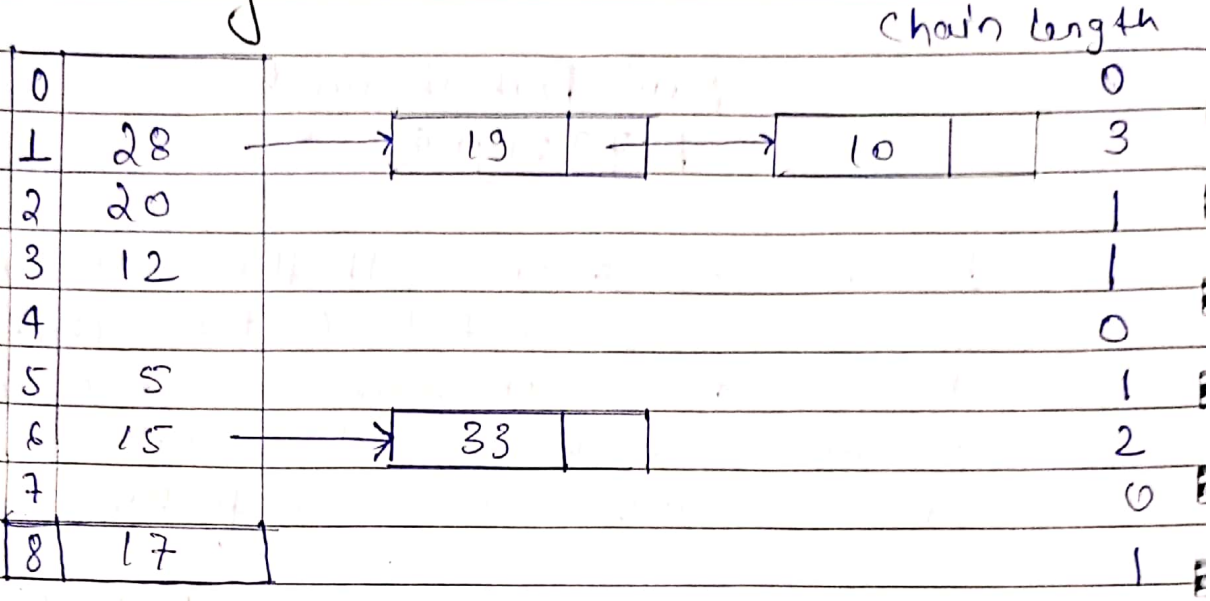
Ans:- h(k) = k mod 9 keys = 5, 28, 19, 15, 20, 33, 12, 17, 10

h(5) = 5 % 9 = 5
h(28) = 28 % 9 = 1
h(19) = 19 % 9 = 1
h(15) = 15 % 9 = 6
h(20) = 20 % 9 = 2
h(33) = 33 % 9 = 6
h(12) = 12 % 9 = 3
h(17) = 17 % 9 = 8
h(10) = 10 % 9 = 1

0	
1	28 → 19 → 10
2	20
3	12
4	
5	5
6	15 → 33
7	
8	17

Ravi Prakash singh
R177219145

(ii) calculate the maximum, minimum and average chain length in the hash table.

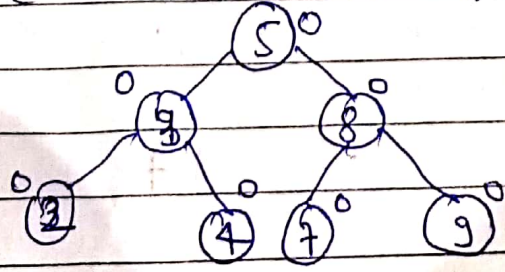


Maximum chain length = 3
Minimum " " = 0
Average = $\frac{0+3+1+1+0+1+2+0+1}{9} = \frac{9}{9} = 1$

(4) Explain the properties of an AVL tree. with suitable diagram and explanations insert the element 21, 26, 30, 9, 4, 14, 28, 18, 15, 10, 2, 3, 7 AVL tree by performing the necessary rotation.

Ans:- Properties of AVL trees:-

(1) it must follow the property of BST
(2) it has only balance factor 0 or +1, or -1 and it cannot be violated.



here Balance factor of all node is zero

Ravi Prakash singh
R177219145.

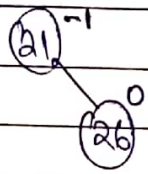
__/__/____

The above example also follows the property of BST i.e. left node always contains small element than root and right node contains greater element

The element that has to be inserted are:- 21, 26, 30, 9, 4, 14, 28, 18, 15, 10, 2, 7

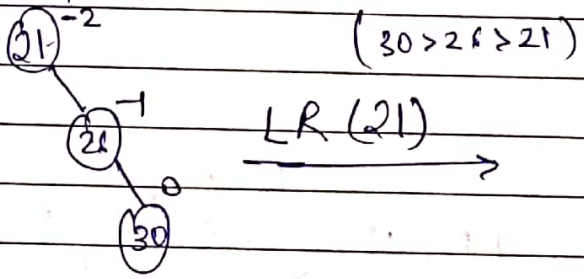
Insert 21

Insert 26



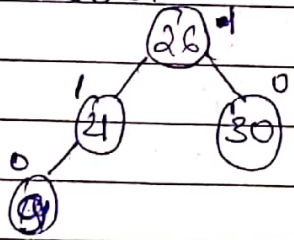
(26 > 21)

Insert 30

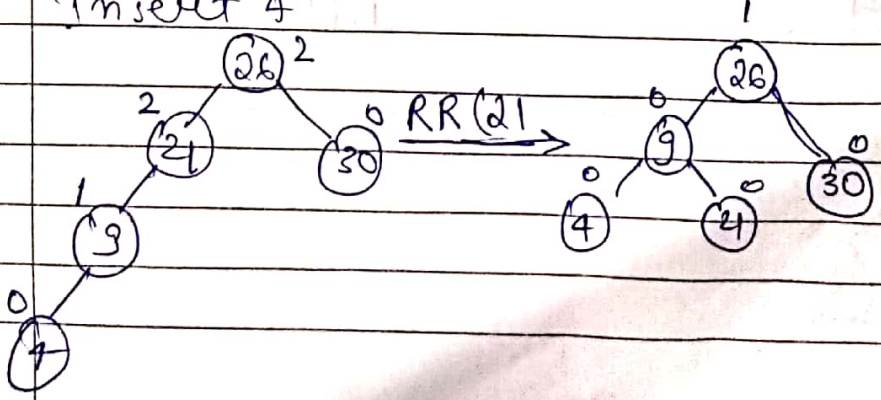


(30 > 26 > 21)

Insert 9

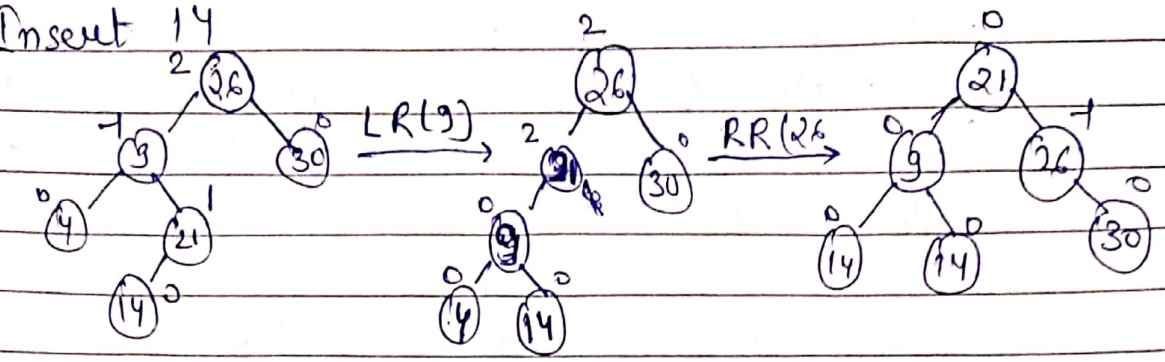


Insert 4

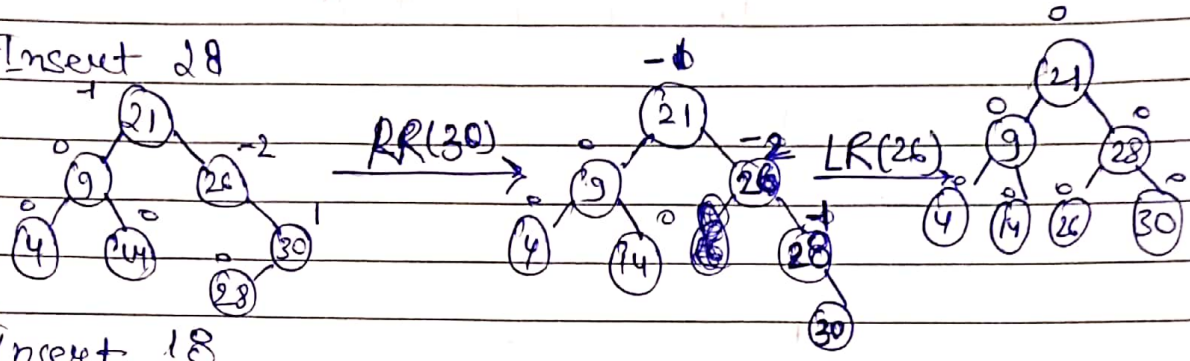


Ravi Prakash singh
R177219145 __/__/____

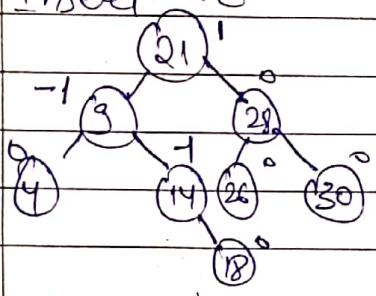
Insert 14



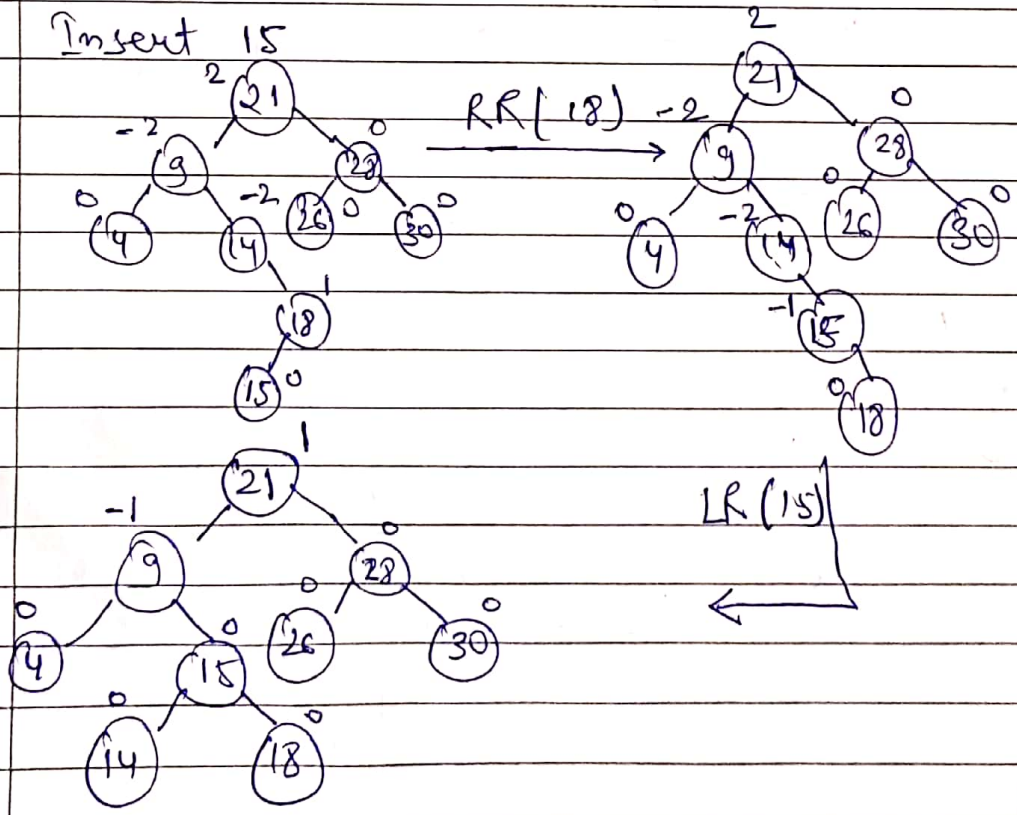
Insert 28



Insert 18



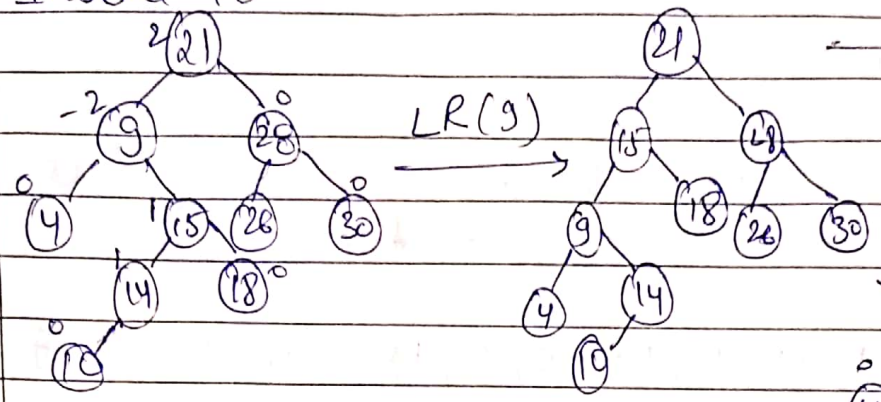
Insert 15

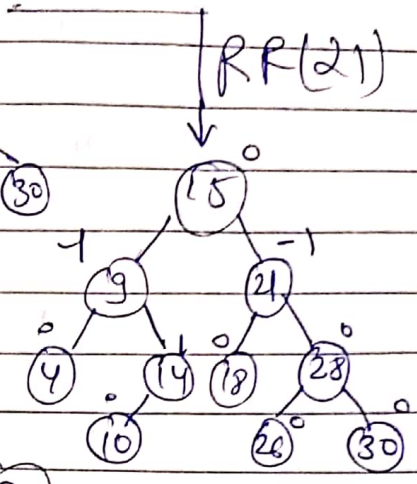


Ravi Prakash singh
R177219145

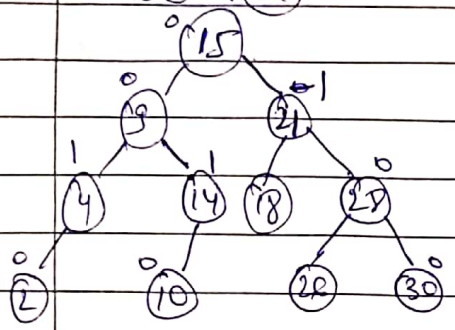
__/__/____

Insert 10

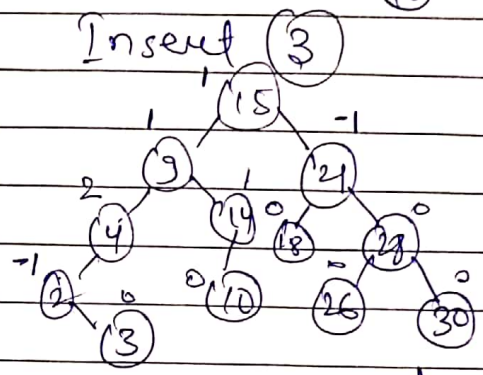


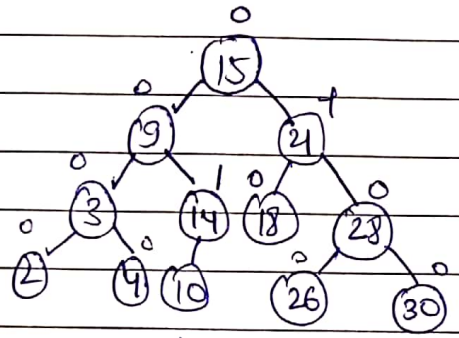


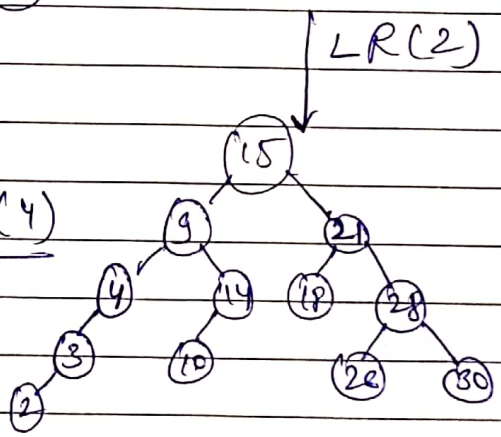
Insert (2)



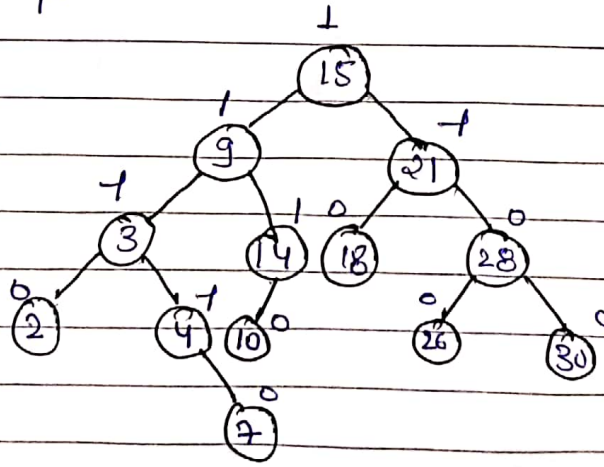
Insert (3)







Insert 7



Ravi Prakash Singh
R177219145

__/__/____

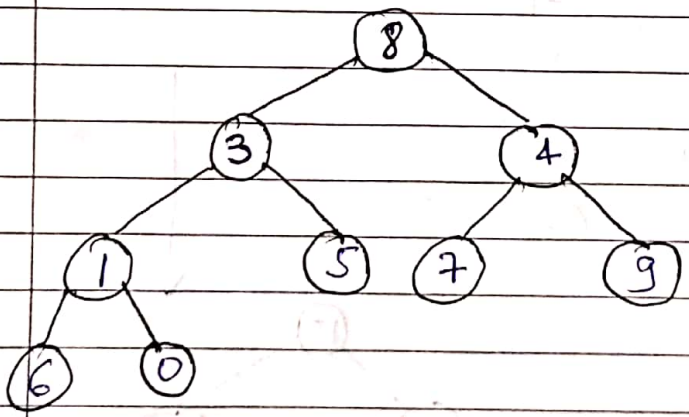
(5) Explain the properties of a Binary Heap tree. the Breadth-first traversal pattern of a Complete Binary tree is 8, 3, 4, 1, 5, 7, 9, 6, 0. construct a CBT and later sort the element in the CBT wing Heap sort algorithms to obtain the elementary in Desecting order

Ans properties of Binary Heap tree

(i) ordering :- Nodes must be arranged in an order according to values. the values should follow min-heap or max-heap property.

(ii) Structural :- All levels in a a heap should be full. we can say that it should be a complete Binary tree

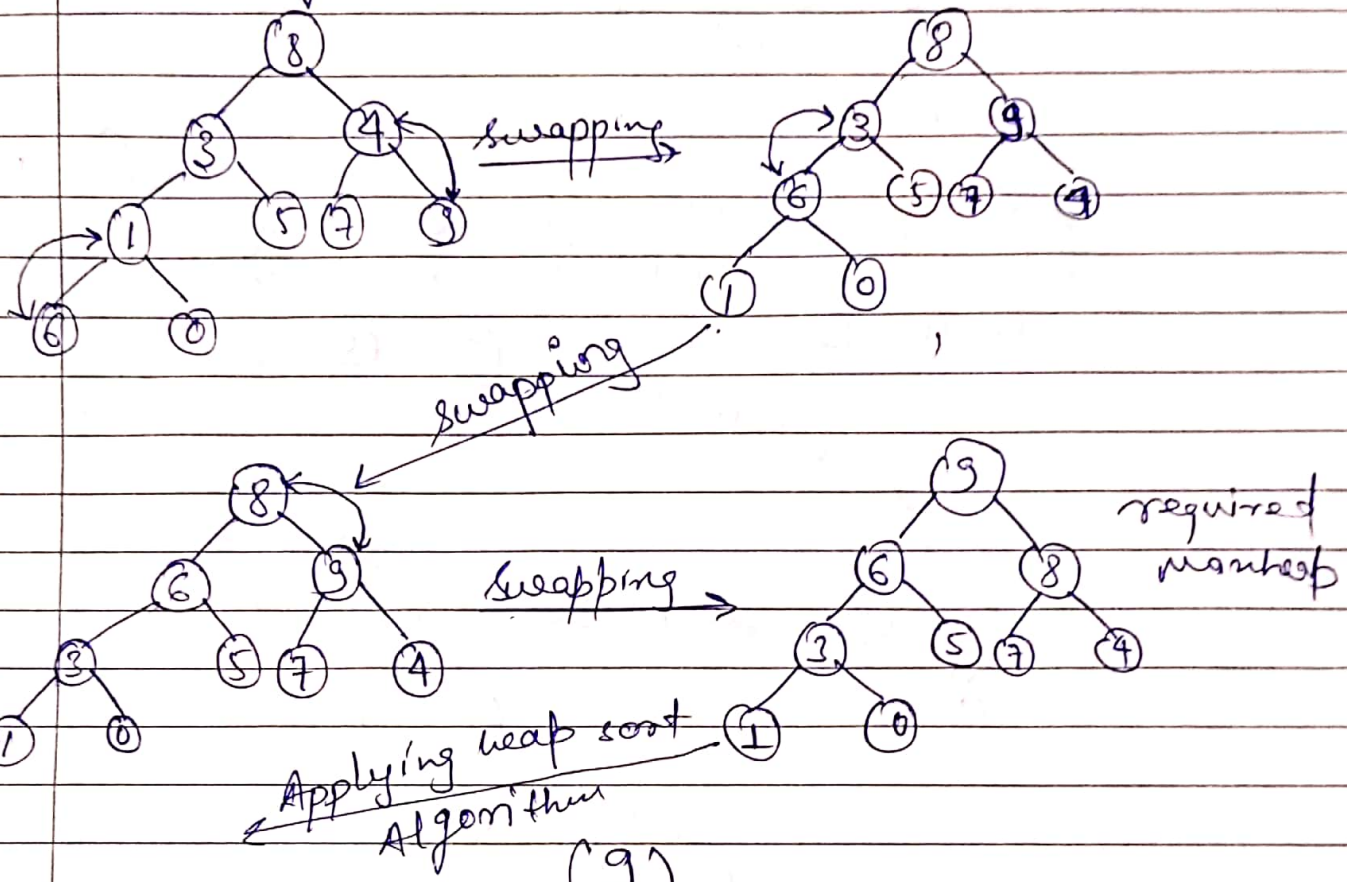
CBT according to the BFT pattern



Ravi Prakash singh
R177219145

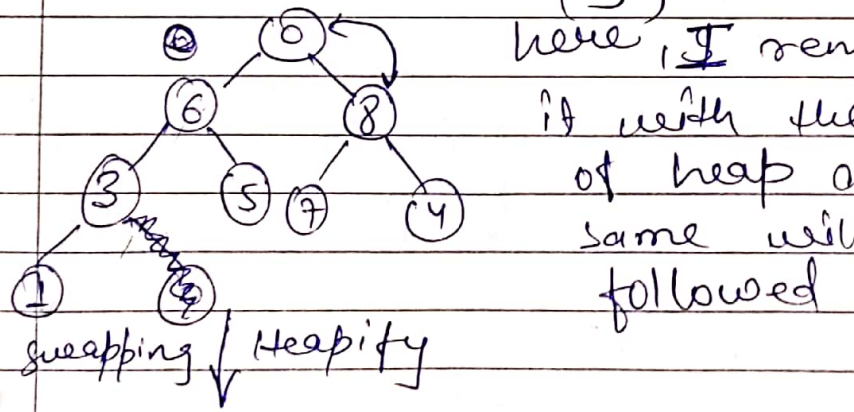
__/__/____

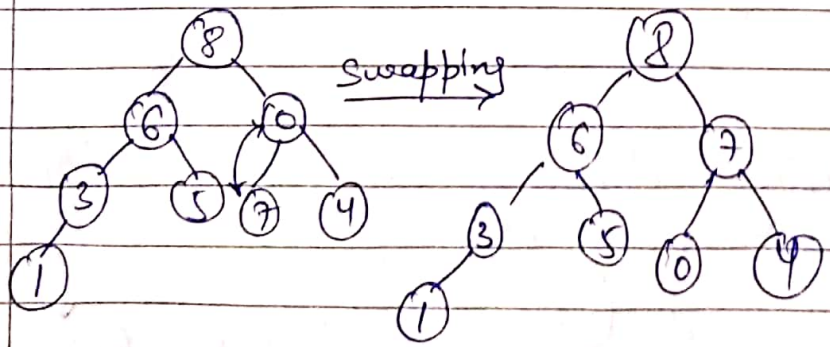
Constructing Max-heap to apply Heap sort algorithm to obtain the elements in Descending order



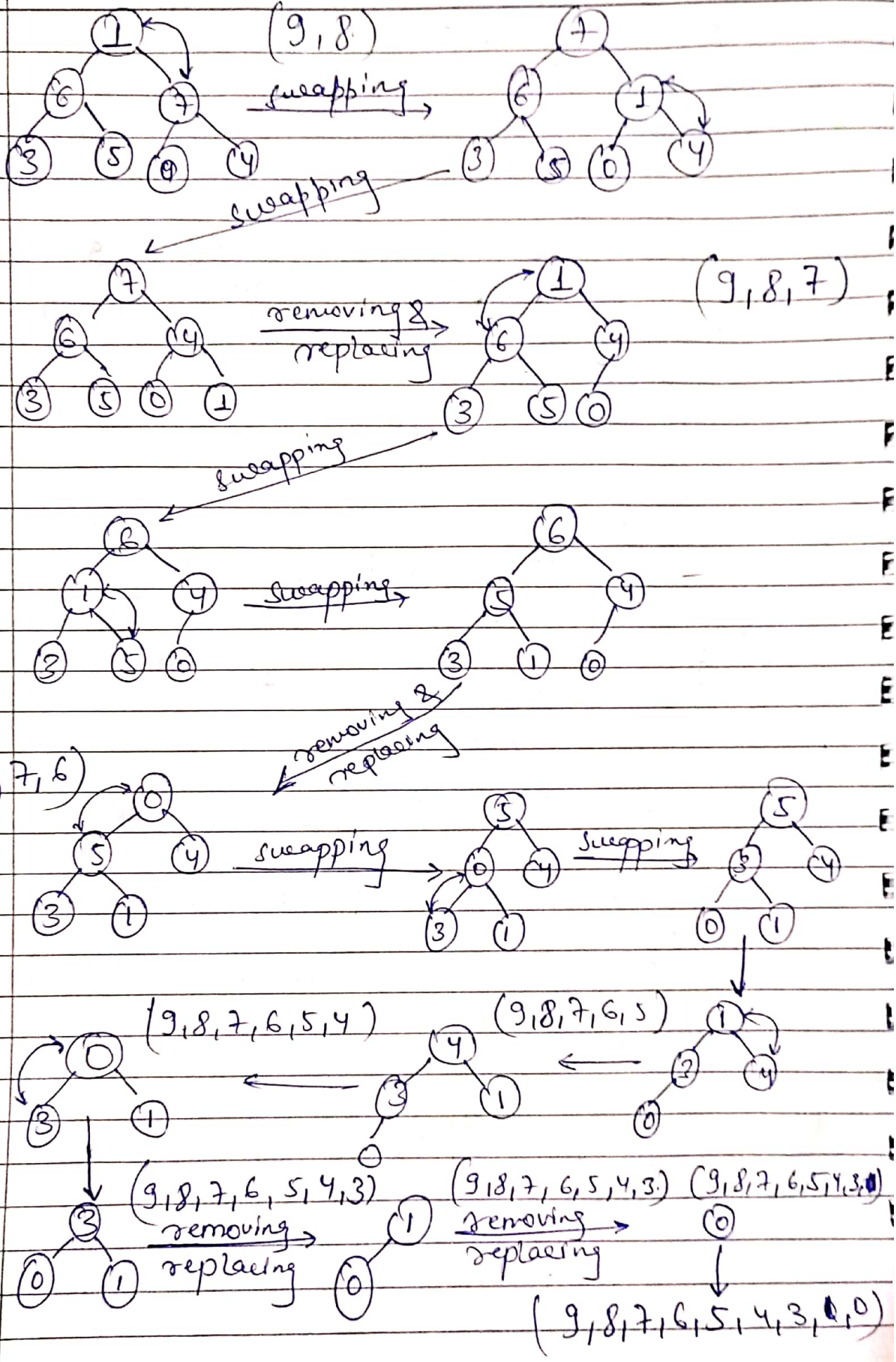
(9)

here, I removed 9 by replacing it with the last element of heap and In all ocases same will be repeated followed by heapify.





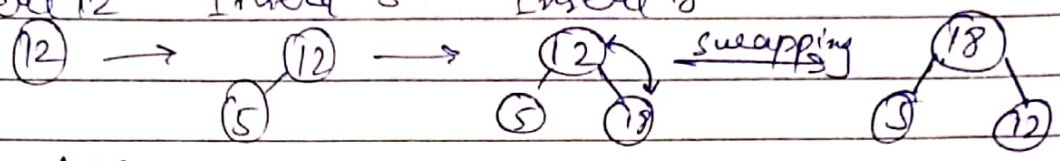
Ravi Prakash Singh
R177219145



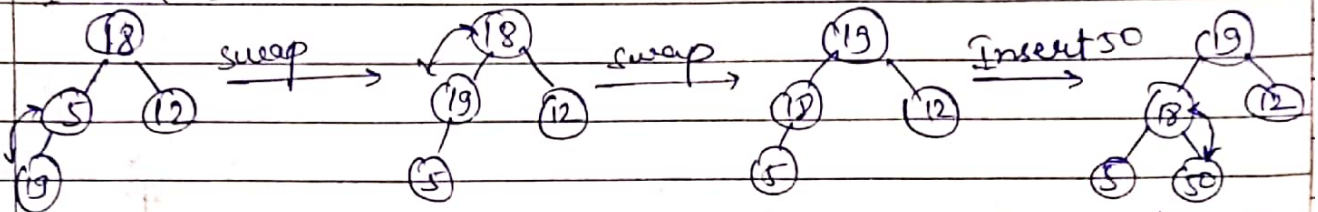
Ravi Prakash Singh
R177219145 __/__/____

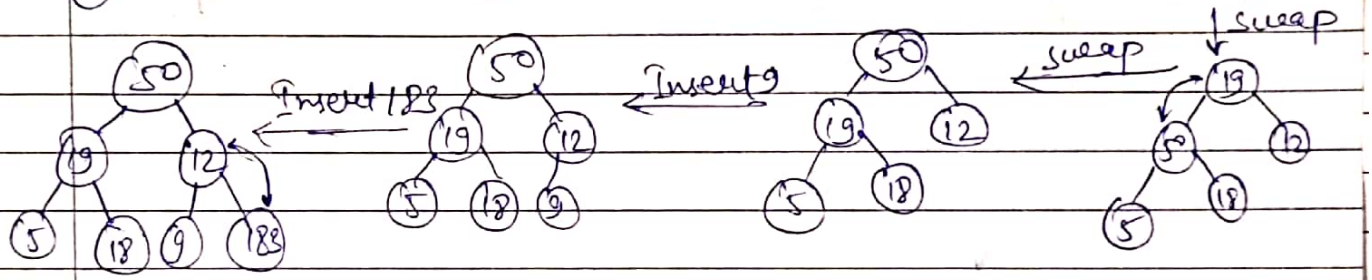
Q6 Create Max-heap and min-heap from the given data and sort in descending and ascending order respectively.
(12, 5, 18, 19, 50, 9, 18, 3, 17, 34, 54, 73, 64, 10)

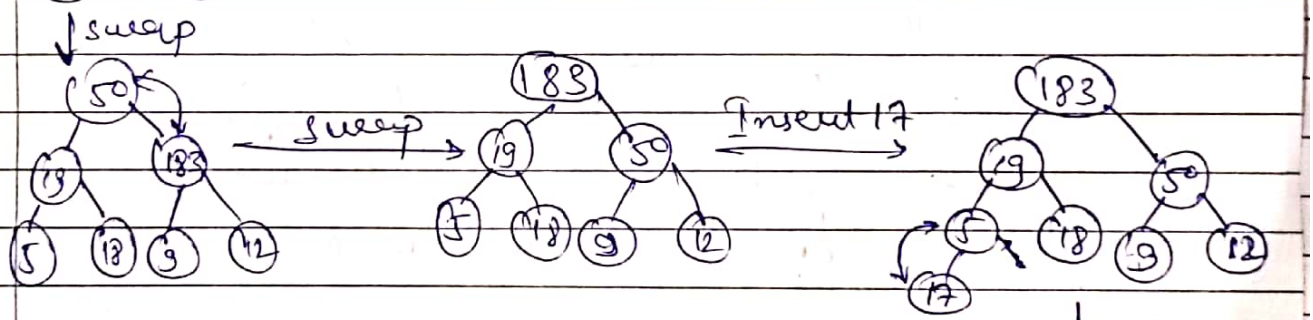
Insert 12 Insert 5 Insert 18

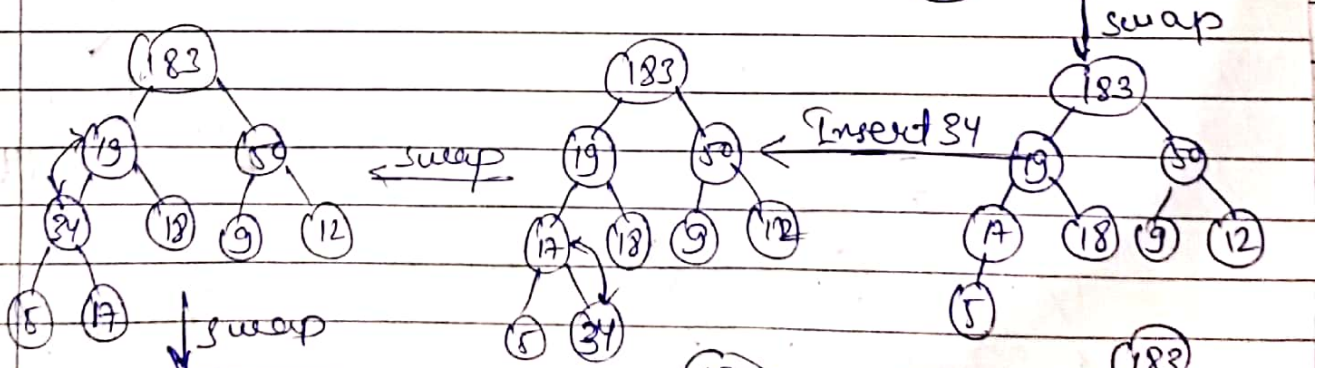


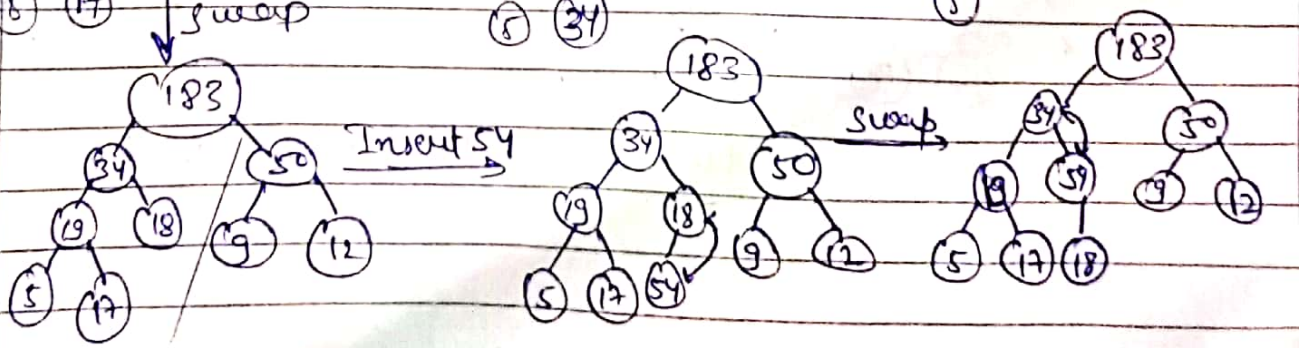
Insert 19



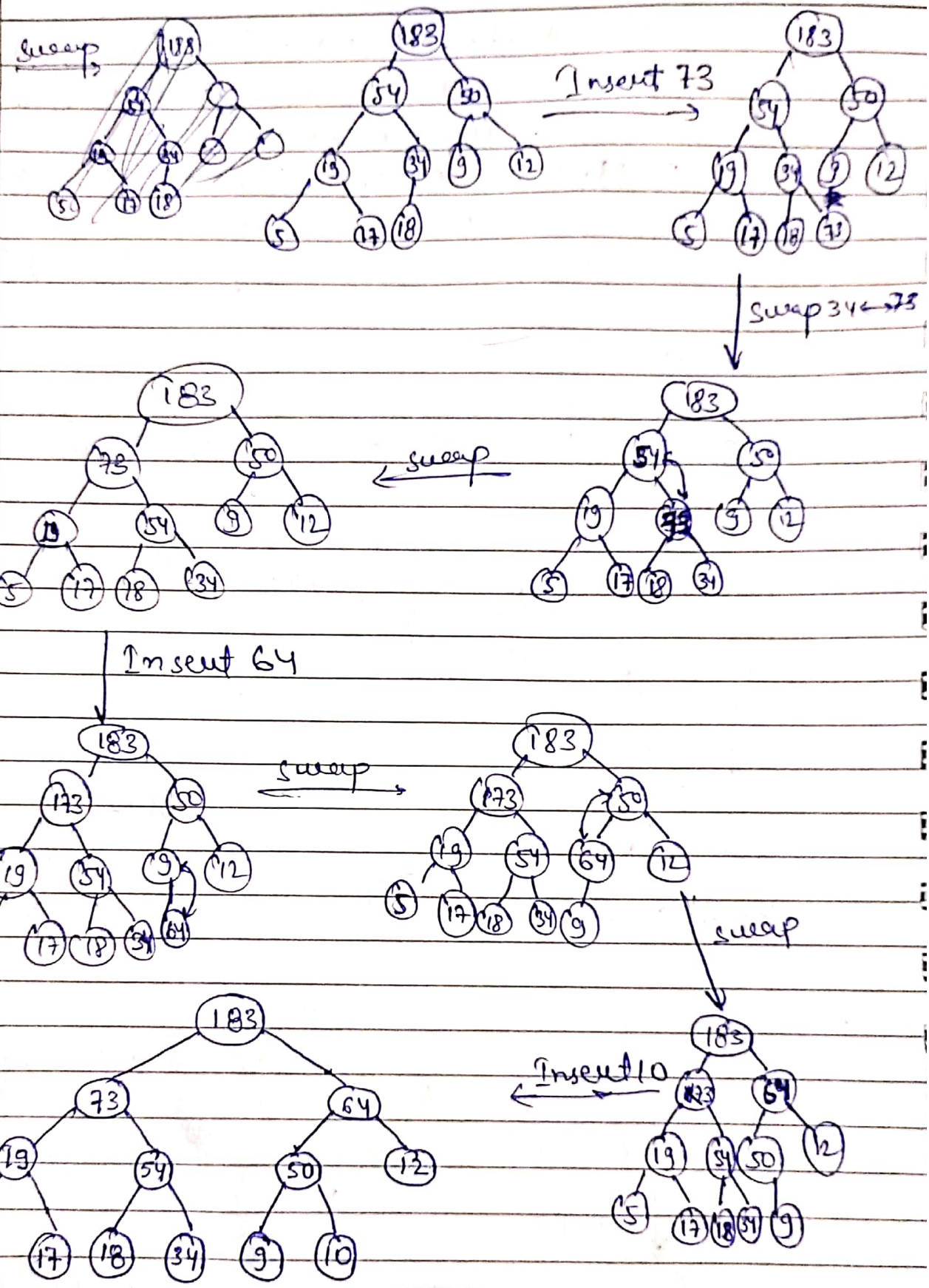








Ravi Prakash singh
R177219145

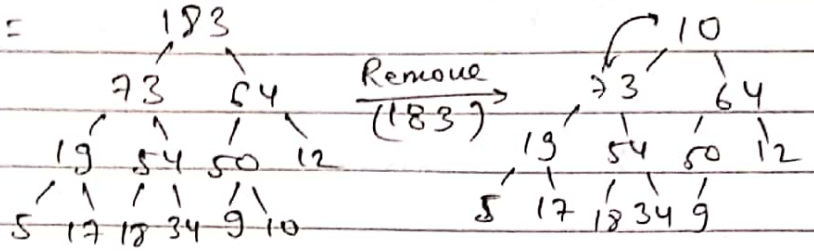


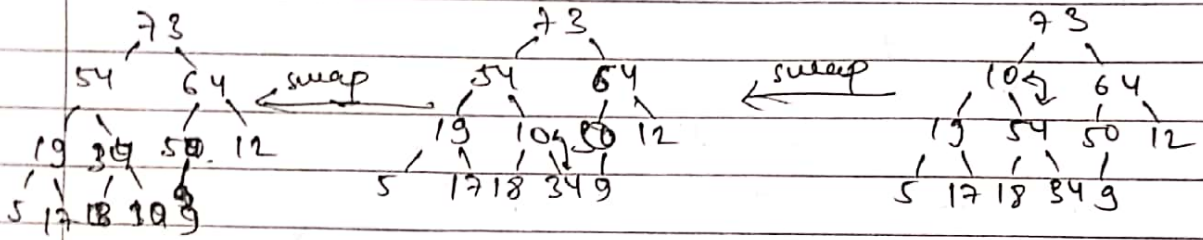
This is the required max heap.

Ravi Prakash singh
R177219145

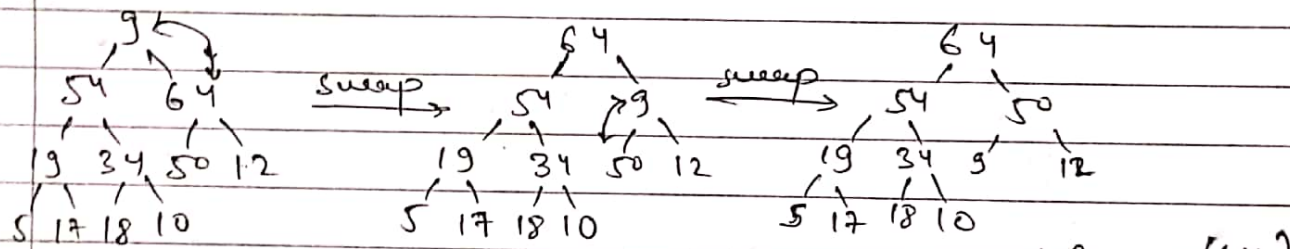
Sorting the max-heap in Descending order

Max-heap =

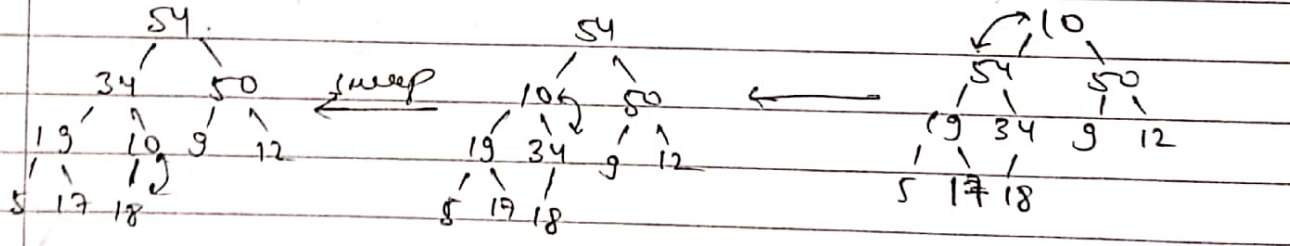




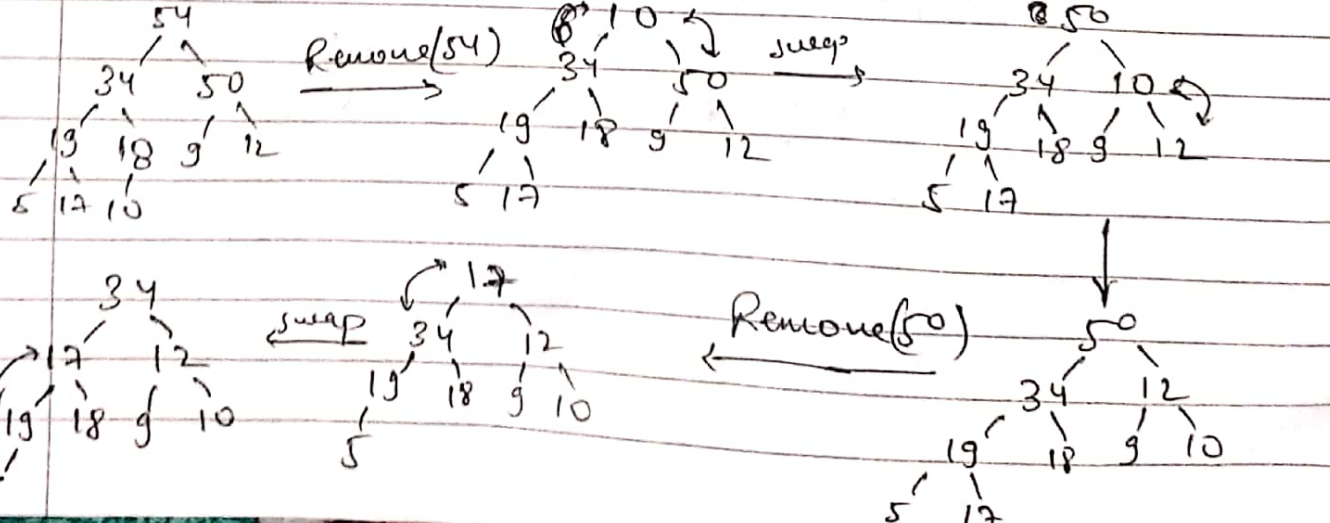
Remove(73)



Remove(64)

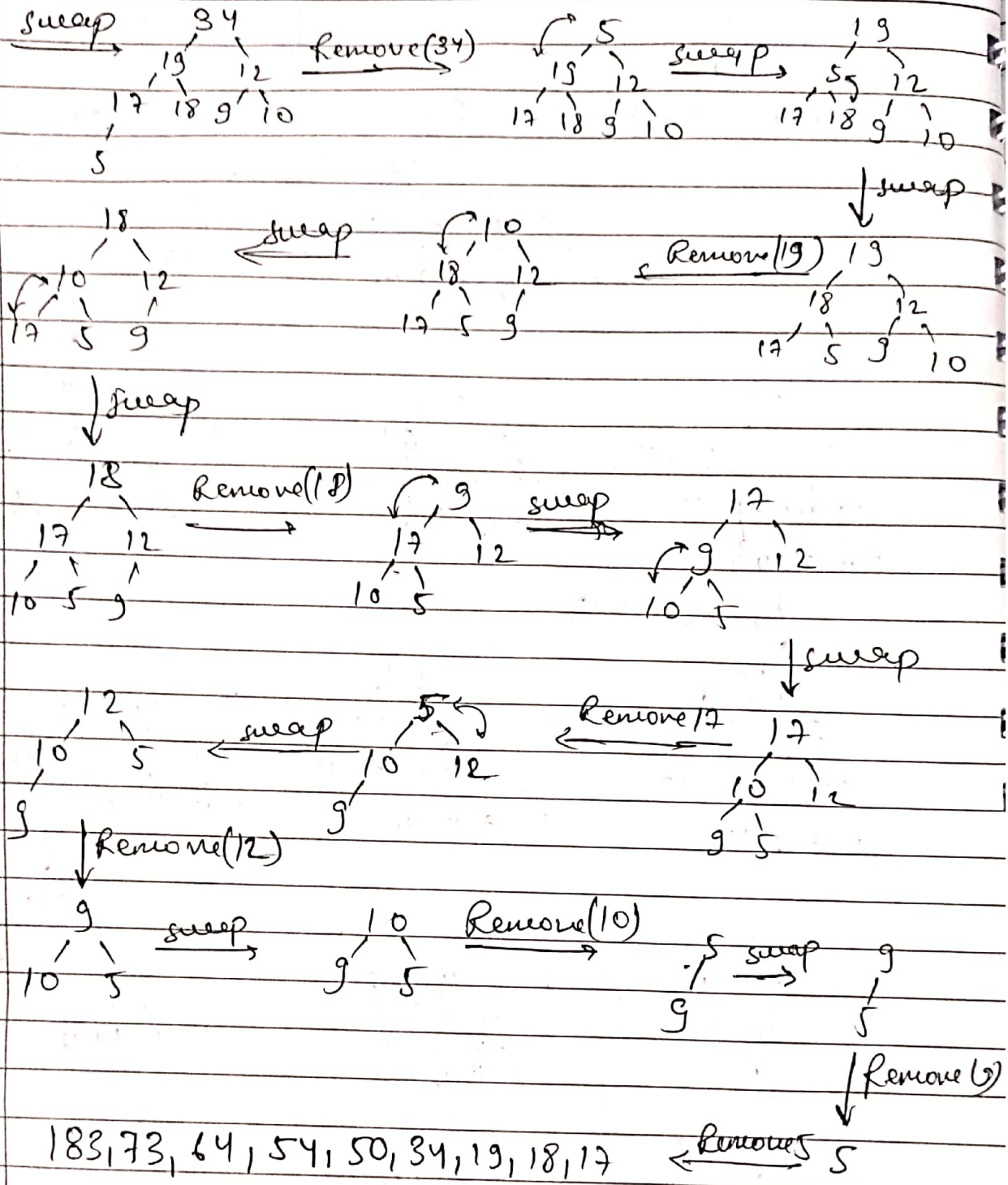


swap



Ravi Prakash singh
R177219145

__/__/____



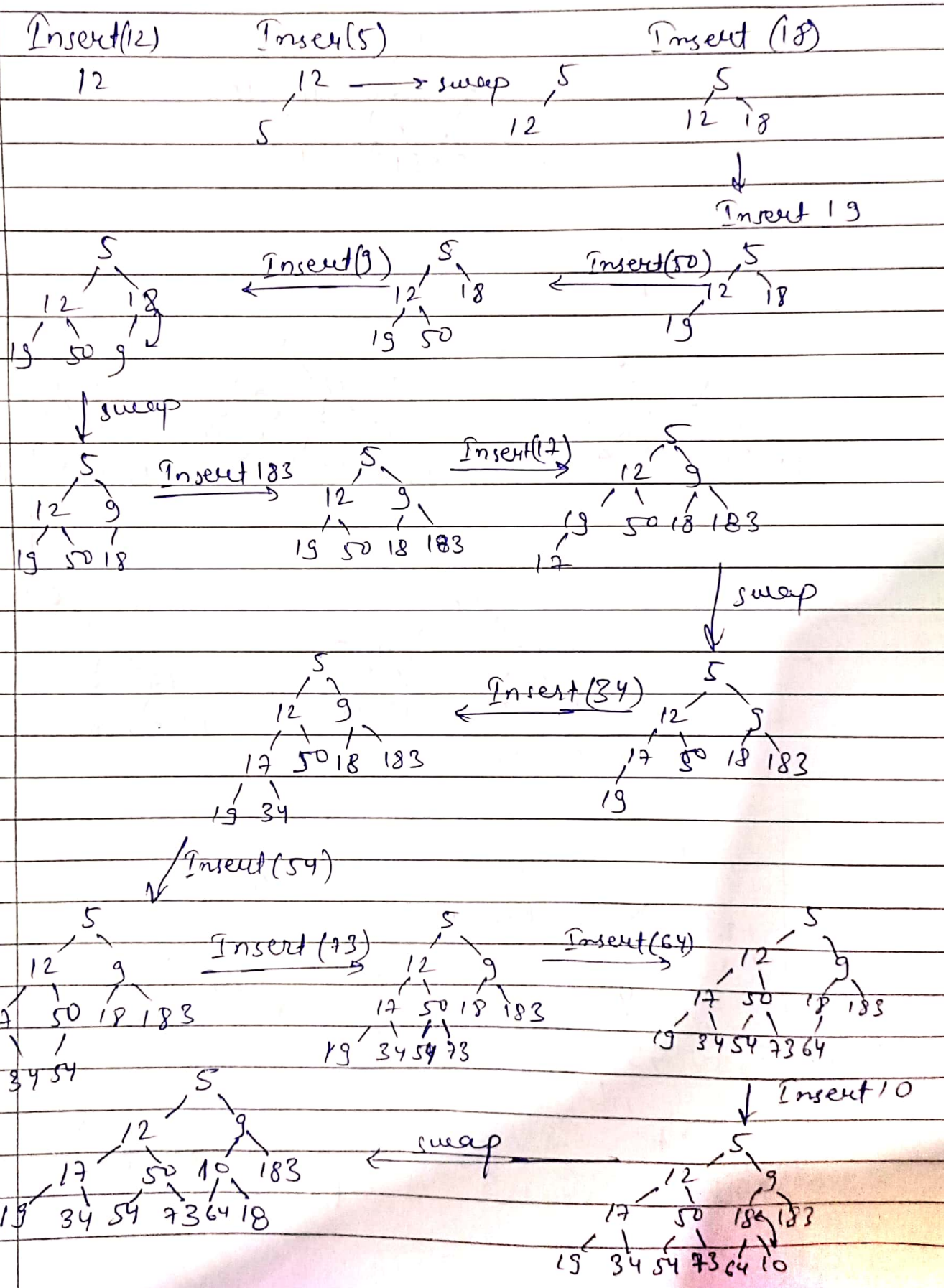
183, 73, 64, 54, 50, 34, 19, 18, 17, 12, 10, 9, 5

we get descending order

Ravi Prakash singh
R177219145

__/__/____

Now creating min-heap with given data.
(12, 5, 18, 19, 50, 9, 183, 17, 34, 54, 73, 64, 10)

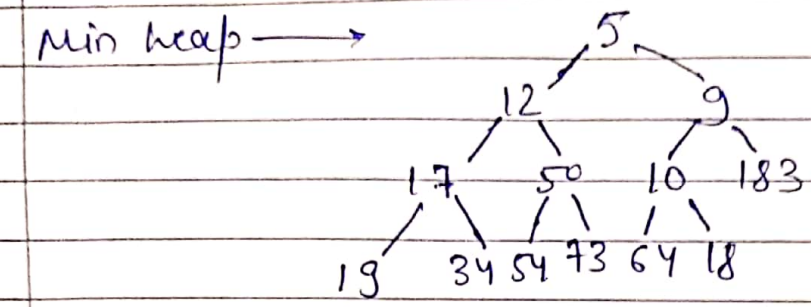


Ravi Prakash Singh
R177219145

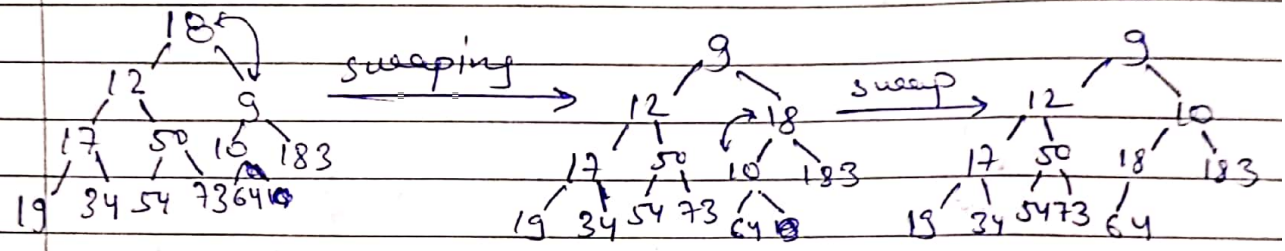
__/__/____

Sorting the min heap to ascending order.

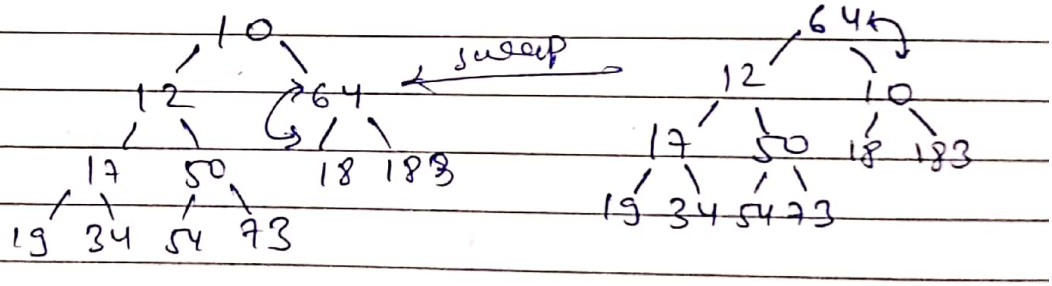
Min heap →



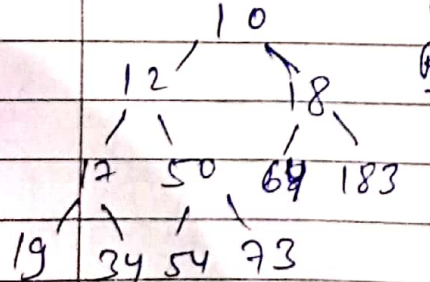
Removing (5)



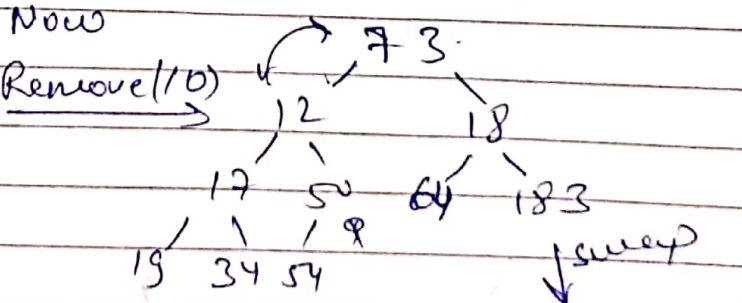
remove (9)

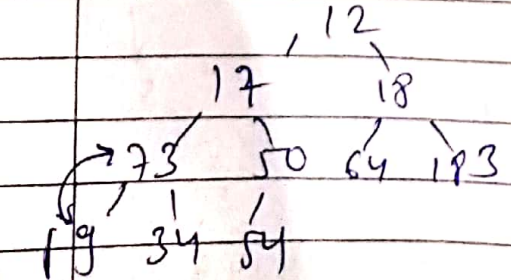


swap

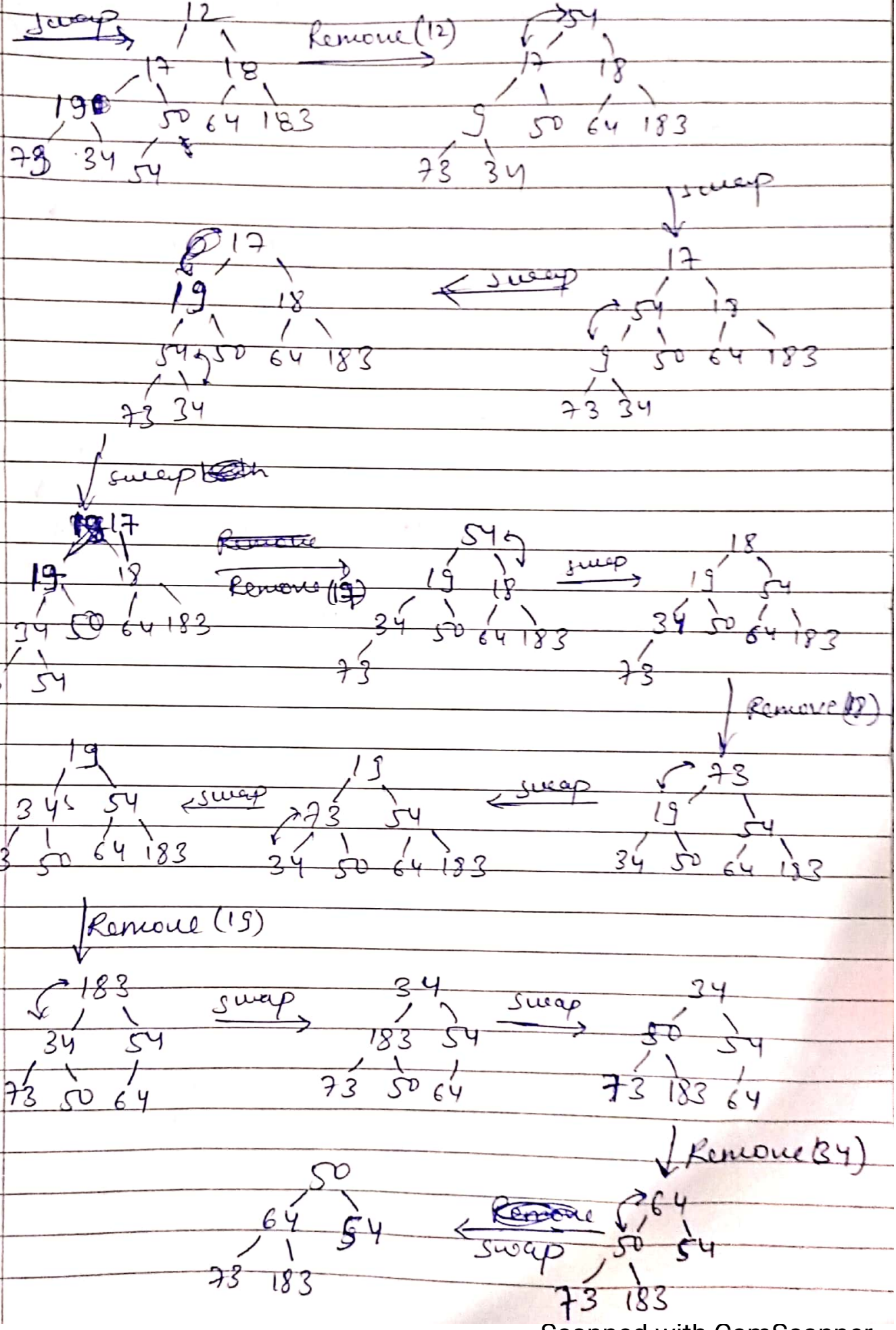


Now Remove (10)

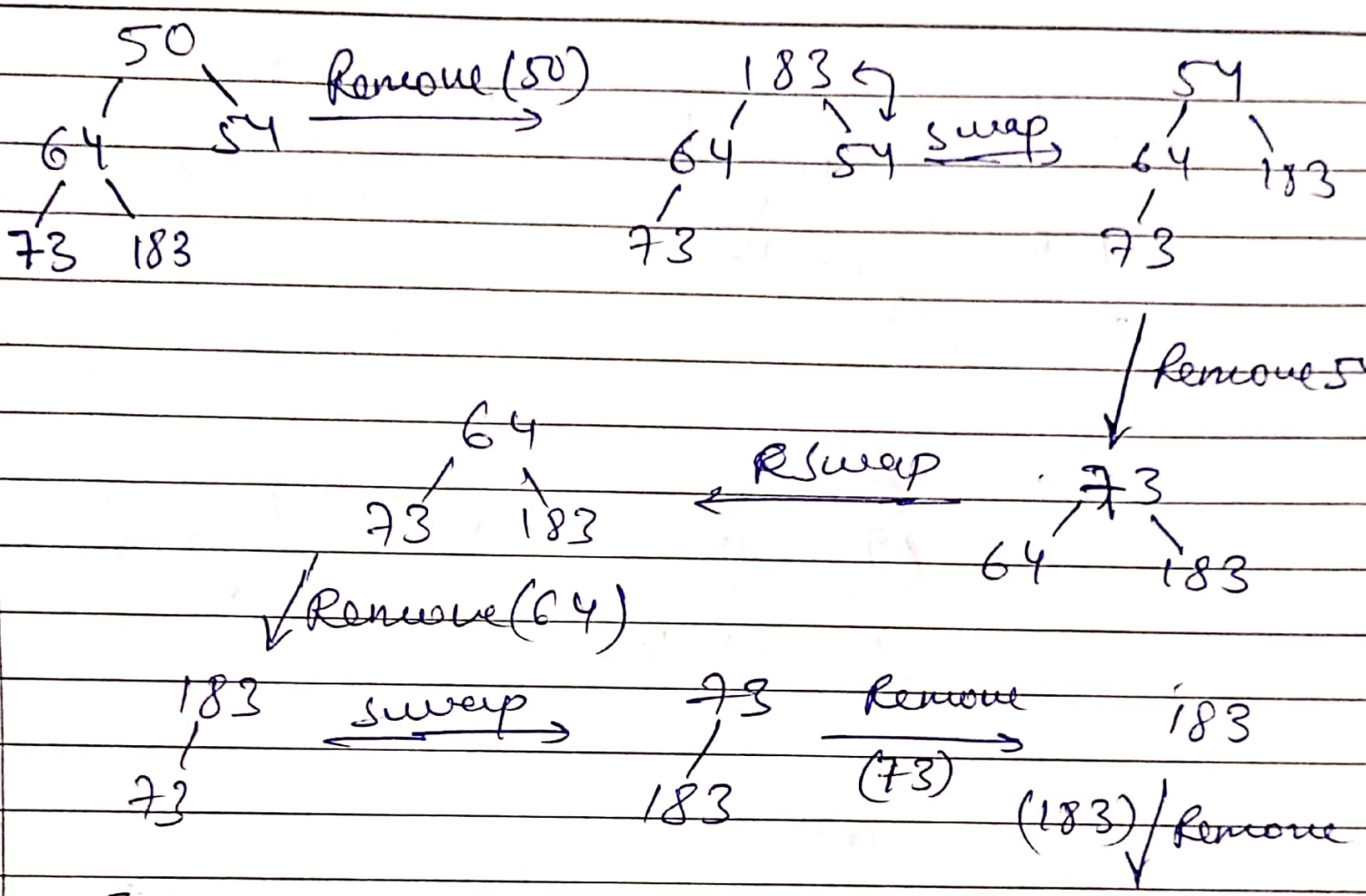




Ravi Prakash singh
R177219145



Ravi Prakash singh
R177219195



5, 9, 10, 12, 17, 18, 19, 34, 50, 54, 64, 73, 183
we get the elements in ascending order.

Ravi Prakash singh
R177219145

7) Find the k^{th} largest element in an array using Heap sort

```
#include<stdio.h>
#include<iostream.h>
using namespace std;

void swap(int *a, int *b)
{
    *a = *a + *b;
    *b = *a - *b;
    *a = *a - *b;
}
void minheapify(int a[], int size, int i)
{
    int l = 2*i;
    int r = 2*i+1;
    if (l<size && a[l] < a[smallest])
        smallest = l;
    if (r<size && a[r]<a[smallest])
        smallest=r;
    if (smallest != r)
    {
      swap(&a[i], &a[smallest]);
      minheapify(a, size, smallest);
    }
}
void buildMinheap(int a[], int size){
        for(int i= size/2; i >=0 ; i--)
          Minheapify(a,size,i);
}
```

Ravi Prakash singh
R177219145

```
int Kthlargest (int a[], int q[], int size, int k)
{
    int minheap[k];
    int i;
    for(i=0; i<k; i++)
        minheap[i] = a[i];
    Buildminheap(minheap, k);
    for(i=k; i<size; i++)
    {
        if(a[i] > minheap[0])
        {
            minheap[0] = a[i];
            minheapify(minheap, k, 0);
        }
    }
    return minheap[0];
}
int main()
{
    int k, n;
    cout<<"enter the size of array:" <<endL;
    Cin >> n;
    int q[n];
    printf("enter the kth element to search in the
                    position of
                    array:");
    cin >>k;
    cout<<"enter the elements of array:"<<endL;
    for(int j=0; j<n; j++)
    {
        cin >> q[j];
    }
    int size = sizeof(q)/size of(q[0]);
```

Ravi prakash singh
R177219145

```
    cout<<" "<<kthlargest(a,size,k))<<endl;
    return 0;
}
```

Output:

```
Enter the size of array:
5
Enter the kth position of element to search
in the array:2
Enter the elements of the array:
5
24
6
8
13
the 2 th largest element is = 13
```