Data Structure and Advanced Programming

(Programming) Homework #9
Due: 2021/5/18 08:00am (CST)

NOTE: Please upload your C++ source codes (by copy-paste) to PDOGS before the due date and time.

1. (30%) Implement a binary search tree. Beginning with an empty tree, insert the input values in the order given, and then print out the tree with preorder, inorder, postorder traversals, respectively. In addition, please also print out the numbers of nodes and leave nodes, and the height of the tree, respectively. After printing out all of the above information, remove the first half of the input file (using round down when the number is aliquant) and print out all information again. When removing a node with two children, please use its inorder successor as the entry.

   Sample Input / Output
   Each input number separating by space of the file is the value to be inserted to the ee. Please use the sample output format to show the results.

```
// input
1

// output
Preorder: 1
Inorder: 1
Postorder: 1
Number of nodes: 1
Number of leave nodes: 1
Height: 1
** after removing 0 nodes **
Preorder: 1
Inorder: 1
Postorder: 1
Number of nodes: 1
Number of leave nodes: 1
Height: 1
```

```
// input
1 3 5 2 4 6

// output
Preorder: 1 3 2 5 4 6
Inorder: 1 2 3 4 5 6
Postorder: 2 4 6 5 3 1
Number of nodes: 6
Number of leave nodes: 3
Height: 4
** after removing 3 nodes **
Preorder: 4 2 6
Inorder: 2 4 6
Postorder: 2 6 4
Number of nodes: 3
Number of leave nodes: 2
Height: 2
```

2. (30%) Implement a minheap. Beginning with an empty heap, insert the input values in the order given, and then print out the heap with preorder, inorder, postorder traversals, respectively. In addition, please also print out the numbers of nodes and leave nodes, and the height of the heap, respectively. After printing out all of the above information, remove the first half of the smallest numbers (using round down when the number is aliquant) and print out all information again.

   Sample Input / Output
   Each input number separating by space of the file is the value to be inserted to the heap. Please use the sample output format to show the results.

```
// input
1
```

```
// input
1 3 5 2 4 6
```

```
// output
Preorder: 1
Inorder: 1
Postorder: 1
Number of nodes: 1
Number of leave nodes: 1
Height: 1
** after removing 0 nodes **
Preorder: 1
Inorder: 1
Postorder: 1
Number of nodes: 1
Number of leave nodes: 1
Height: 1
```

```
// output
Preorder: 1 2 3 4 5 6
Inorder: 3 2 4 1 6 5
Postorder: 3 4 2 6 5 1
Number of nodes: 6
Number of leave nodes: 3
Height: 3
** after removing 3 nodes **
Preorder: 4 6 5
Inorder: 6 4 5
Postorder: 6 5 4
Number of nodes: 3
Number of leave nodes: 2
Height: 2
```

3. (40%) Any sequence *A* of size *n* is called **B-sequence** if: $A_1 < A_2 < ... < A_k > A_{k+1} > A_{k+2} > ... > A_n$ where $1 <= k <= n$. That is, a sequence which is initially strictly increasing and then strictly decreasing (the decreasing part may or may not be there). All elements in *A* except the maximum element comes at most twice (once in increasing part and once in decreasing part) and maximum element comes exactly once. All elements coming in decreasing part of sequence should have come once in the increasing part of sequence. You are given a B-sequence *S* and *q* operations. For each operation, you are given a value *v*. You have to insert *v* in *S* if and only if after insertion, *S* still remains a B-sequence. After each operation, print the size of *S*. After all the operations, print the sequence *S*.

   Sample Input / Output
   The first input line consists of an integer N, denoting size of S, the second line consists of N space separated integers, denoting elements of S, and the next line consists of an integer q, denoting number of operations. Each of the following q lines consists of an integer v. After each operation, print the size of S in a new line. After all operations, print the sequence S.

```
// input
4
1 2 5 2
4
5
1
3
2

// output
4
5
6
6
1 2 3 5 2 1
```

```
// input
2
1 2
4
2
1
3
2

// output
2
3
4
5
1 2 3 2 1
```