Indian Institute of technology, Guwahati
Department of Computer Science and Engineering
Data Structure Lab: (CS210)
Lab Assignment: 4

Date: 9th October, 2017 **Total Marks: 20**

1. A ternary tree is a rooted tree in which each node can have three children (left, middle, and right). A ternary min-heap is an (almost) complete ternary tree in which the value stored at any node is no larger than the values stored at its child nodes. Assume that each node stores a positive integer value. Store your ternary min-heap in a static array of integers **H[100]**. Maintain its size n separately. (n is less than or equal to 100).
   Write the ternary min-heap functions with the following prototypes:

   void **buildheap**(int H[], int n) /*covert array H[] to a ternary min-heap*/ **[8]**
   void **insert**(int H[], int n, int x) /*Insert x to a heap H[] of size n*/ **[6]**
   int **deletemin**(int H[], int n) /* The return value is the deleted minimum*/ **[6]**

   Use a command interface to perform insert and deleteMin based on user input. Use I for insert and D for deleteMin.

   Input : First line will contain an integer **n** denoting number of elements in array.
   Second line will contain n space separated positive integers.
   Following lines will contain either **"I followed by an integer x"** for inserting x in the heap or **"D"** for deleteMin operation.

   Output: First line will be min-heap formed by input array elements.
   Following lines will depend on the operation specified by user.
   - If user has given command to insert a number x. Print "x is inserted" and the updated heap after insertion in the next line.
   - If user has given command to deletemin & suppose minimum number is a then print "a is deleted" and updated heap in the next line.

   Test Case:
   Input:

   80
   23 6 57 35 33 15 26 12 9 61 42 27 50 59 3 6 60 66 52 56 11 8 7 29 22 10 62 3 67 15 9 42 22 18 29 7 33 56 51 42 69 13 21 39 24 57 78 4 75 50 13 6 11 20 36 33 62 50 36 1 65 45 44 7 16 25 46 49 33 17 44 55 62 65 14 7 74 44 43 70

   I 3
   D
   I 10
   I 1
   D
   I 8
   I 53

Output:

1 3 3 7 4 6 6 7 9 9 18 27 42 13 24 6 13 11 33 26 7 8 12 29 7 10 62 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 23 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70

3 is inserted

1 3 3 7 4 6 6 7 3 9 18 27 42 13 24 6 13 11 33 26 7 8 12 29 7 10 9 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 23 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70 62

1 is deleted
3 4 3 7 6 6 6 7 3 9 18 27 42 13 24 23 13 11 33 26 7 8 12 29 7 10 9 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 62 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70

10 is inserted
3 4 3 7 6 6 6 7 3 9 18 27 42 13 24 23 13 11 33 26 7 8 12 29 7 10 9 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 62 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70 10

1 is inserted

1 4 3 7 6 6 6 7 3 9 18 27 42 13 24 23 13 11 33 26 7 8 12 29 7 10 3 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 62 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70 10 9

1 is deleted
3 4 3 7 6 6 6 7 3 9 18 27 42 13 24 23 13 11 33 26 7 8 12 29 7 10 9 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 62 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70 10

8 is inserted
3 4 3 7 6 6 6 7 3 9 18 27 42 13 24 23 13 11 33 26 7 8 12 29 7 10 8 57 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 62 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70 10 9

53 is inserted
3 4 3 7 6 6 6 7 3 9 18 27 42 13 24 23 13 11 33 26 7 8 12 29 7 10 8 53 67 15 61 42 22 42 29 35 33 56 51 50 69 59 21 39 33 57 78 62 75 50 15 60 66 20 36 52 62 50 36 56 65 45 44 11 16 25 46 49 33 17 44 55 62 65 14 22 74 44 43 70 10 9 57