

Indian Institute of technology, Guwahati
Department of Computer Science and Engineering
Data Structure Lab: (CS210)
Offline Assignment: 7

Date: 9th October 2017.

Total Marks: 30

Deadline: 10PM, 15th October 2017. (Hard Deadline)

1. [In continuation of the Lab Assignment 4] You should use your implementation of LA4 wherever required.

- a. Write a function with the following prototype to print the k smallest elements stored in a ternary min-heap H: **void ksmallest (int H[], int n, int k);** The function should print the elements in sorted (increasing) order, and have a running time of $O(k \log n)$, where n is the number of elements stored in H. A heap is allowed to store the same value at multiple nodes. Your function should print only k values including repetitions. After the function returns, the heap should continue to contain all of its n elements. **[5]**
- b. Write a function with the following prototype to print all elements less than equal to a in a ternary min-heap H: **void printsmall (int H[], int n, int a);** Printing need not be in a sorted order. If some element less than equal to a appears multiple times in H, print it as many times as it occurs. Your function must run in $O(t)$ time, where t is the number of elements printed. The heap should continue to contain all of its n elements after the function returns. **[5]**

Your main should do the following:

- a) Read the size n of the heap
- b) Read n positive integers and store them in an array H[].
- c) Run buildheap() to convert the array to a ternary min-heap. Print the array H[].
- d) Read k from the user. Call the function **ksmallest** to print the k smallest elements in your heap.
- e) Read a. Invoke the function **printsmall** to print all heap elements less than or equal to a.

Input: First line containing a number n,size of heap

Second line containing n positive integers.

Third line containing a single integer k.

Fourth line containing a single integer a.

Output: First line containing heap elements after buildheap()

Second line containing k smallest integers in sorted order of the heap

Third line containing all heap elements less than or equal to k

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

10

20

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

1 3 3 4 6 6 6 7 7 7

[Print H here to ensure H is not modified by the above operation]

1 3 3 4 6 6 6 7 7 7 7 8 9 9 10 11 11 12 13 13 14 15 15 16 17 18 20

[Print H here to ensure H is not modified by the above operation]

2. Implement Fibonacci Heaps. You need to only implement following operations: [20]

- MAKE-FIB-HEAP
- FIB-HEAP-INSERT
- FIB-HEAP-EXTRACT-MIN
- FIB-HEAP-UNION

You can use the command interpreter that uses codes **c**, **i**, **e**, **u** and **S** to make, insert, extract the minimum, do a union operation and to show the heap structure, respectively.

Use test1.txt and test2.txt to test your code.