BPDC, Dubai – II Semester, 2020-2021

Course No: CS F211 **LAB EXERCISE 10** **Course Title: DSA**

Date: April 12 2021

Write a program that implements the following subroutines and hence the task below:
(a) min-heapify(*A[],n,i*)
//a recursive procedure takes an array of n elements (indexed 1...n) and an index i<n as input and performs at most *depth(i)* swaps (where *depth(i)* is the number of edges in the path from node *i* to the farthest leaf in the subtree rooted at *i)* to convert the subtree rooted at *i* to a min-heap. Further, *min-heapify()* assumes both the left and right subtrees of *i* to be min-heaps.

(b) build-heap(*A[],n*)
//a procedure that converts an arbitrary input array to a min-heap (bottom-up) using the above procedure *min-heapify()* as a subroutine in *O(n)* time.

(c) sort(*A[],n*)
//sorts the numbers in descending order using the above *min-heap()* procedure.

```
sort(A[],n)
        build-heap(A[],n)
        for(i← 1 to n-1)
                swap(A[1],A[n+1-i])
                min-heapify(A[],n-i,1)
```

(d) printFirst-k-Smallest(*A[],n,k*)
//Prints the first *k* smallest elements in the list in O(*k* log *n*) time provided the array *A[1...n]* is a min-heap.

```
printFirst-k-Smallest(A[],n,k)
        for(i← 1 to k)
                Print A[1]
                swap(A[1],A[n+1-i])
                min-heapify(A[],n-i,1)
```

As we had before, you are being provided with two files testCaseSize.txt and searchTestCase.txt containing 18 test cases. Your program should read each of those test cases, convert each to a min-heap using your build-heap() procedure and write the resultant heaps to a file *heapLists.txt*. In the meantime, compute the number of pairwise swaps performed by your build-heap() procedure and print the same as output (your program would output 18 values).

Further, use your sort() procedure to sort each of those test cases and write the resultant lists to a file *sortedLists.txt.*

To end with, use your printFirst-k-Smallest() procedure to print the k smallest elements in each of those lists, where k=n%100+r for n being the size of the list, and r being your roll number (the integer composed of last 3 digits). Your procedure should read each of those heaps from the file *heapLists.txt* to begin with, so as to have the input being a min-heap(), thereby satisfying the contraint associated with the function printFirst-k-Smallest(*A[],n,k).*