

Programming Assignment #2

Due at 11:59pm, Tuesday, March 29

All the codes should be written in c or c++ for linux and commented appropriately for major steps/functions

Code that does not compile will not be graded and get a 0 automatically

The codes should be submitted as a single zipped file through Blackboard

Implement the HeapBottomUp Algorithm using C or C++. (100 pts)

```

Algorithm HeapBottomUp( $H[1..n]$ )
//Constructs a heap from the elements of a given array
// by the bottom-up algorithm
//Input: An array  $H[1..n]$  of orderable items
//Output: A heap  $H[1..n]$ 
for  $i \leftarrow \lfloor n/2 \rfloor$  downto 1 do
     $k \leftarrow i$ ;  $v \leftarrow H[k]$ 
    heap  $\leftarrow$  false
    while not heap and  $2 * k \leq n$  do
         $j \leftarrow 2 * k$ 
        if  $j < n$  //there are two children
            if  $H[j] < H[j + 1]$   $j \leftarrow j + 1$ 
        if  $v \geq H[j]$ 
            heap  $\leftarrow$  true
        else  $H[k] \leftarrow H[j]$ ;  $k \leftarrow j$ 
     $H[k] \leftarrow v$ 

```

Requirements:

- You are required to implement a **Max Heap** where the root node contains the largest key
- Your code should be able to read an input ASCII file named 'input.txt', where the first line contains the total number of keys, and the second line contains the unsorted keys (integer numbers) separated by blank space
- Your code will produce an output ASCII file named 'output.txt', which contains the resulted heap $H[1, \dots, n]$ starting from the root, separated by blank space
- Your code should output the execution time for running HeapBottomUp excluding time of input/output.
- **A script file or readme file including the instructions to compile and run the code should be submitted together with the codes**

Extra Credit: Implementing Heapsort Algorithm using C or C++. (20 pts)

Write a C or C++ code to perform Heapsort using your constructed heap. (20 pts)

Requirements:

- Your code should be able to read an input ASCII file named 'input.txt', where the first line contains the total number of keys, and the second line contains the unsorted keys (integer numbers) separated by blank space
- Your code will produce an output ASCII file named 'output.txt', where the first line contains the resulted heap $H[1, \dots, n]$ starting from the root, separated by blank space, and the second line contains the heapsort result starting from the largest number
- A script file or readme file including the instructions to compile and run the code should be submitted together with the codes