# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2021** |
|  | | | | | |
| Points |  | Document author: | **Saishnu Ramesh Kumar** | | |
|  | Author’s email: | **saishnu\_rk@mail.fresnostate.edu** | | |
| Laboratory number: | **Section 1, 11am to 12:50pm** | | |
|  | | | | | |

**1. Statement of Objectives**

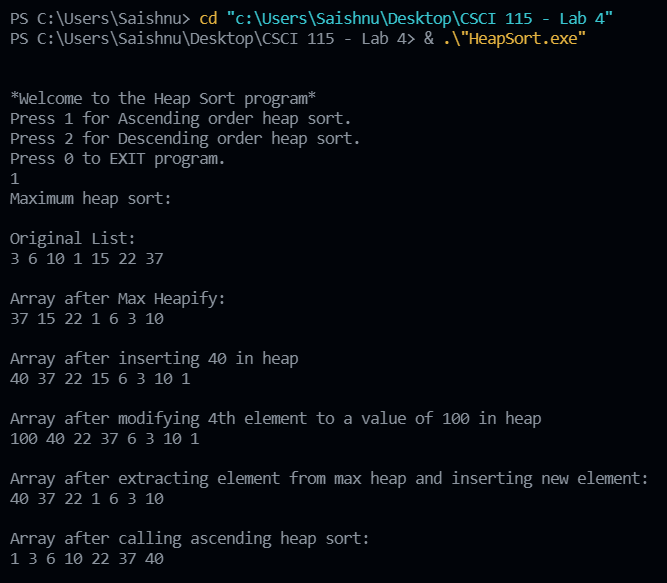
For this lab exercise, we were instructed to create a heap sort function that works both ways. In solving the Max Heap and the Min Heap. This is through the use of arrays that are drawn out on a binary tree therefore we would need to do swapping and several different exercises to create a sorted list. This is essentially another method for sorting because Heap Sort is another form of sorting algorithm. Also, heapsort takes the time complexity of O(nlogn).

**2. Experimental Procedure**

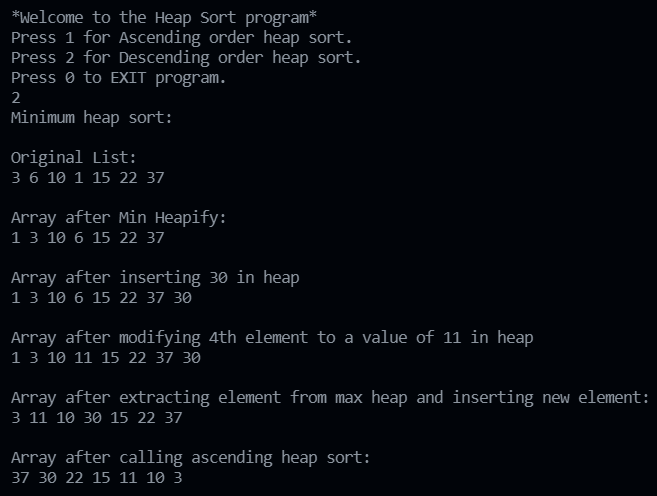
We were given a header file to follow, regarding what functions should be used and what parameters are included in those functions. A Heap class was created that contained all the functions and I have even listed the function prototypes on the top of the program. Within each function, they are given different tasks to do in the program, whether it be the max heap functions or the min-heap functions, they work about the same, but the data inserted may just be the opposite because max heap has the biggest number of the top of the tree whereas min-heap has the smallest number on the top of the tree. I have also made some modifications to the constructor to allow it to run through the whole list of the array as well as checking the size of the list. There is also a swap function, to help with swapping the numbers around, mainly the last element and the first element when you are trying to remove the topmost element in the list. For the main function, I created a switch menu system, therefore the user would be able to pick which type of heap sort they would like. Moreover, I added an exit statement so that they could exit the program whenever they feel like it.

**3. Analysis**

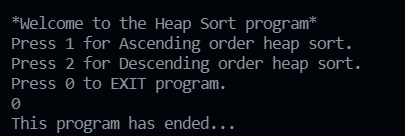
Max Heap Terminal Screenshot:



Min Heap Terminal Screenshot:



Exit Terminal Screenshot:



**4. Encountered Problems**

For this experiment, I encountered numerous compiler errors and was somewhat able to solve them more or less. This exercise was particularly more challenging, but I feel that I was able to go through with it to an extent.

**5. Conclusions**

After completing this lab exercise, I have got a better understanding of how I should be coding heap sort, both the max heap and min heap, because I feel that I have a clear understanding of how it works theoretically but implementing this sorting algorithm was fairly challenging. I think I was able to do it after doing some research online and going through the slides provided during the lab and the lecture was helpful.

**6. References**

1. Slides from the lab, provided by the TA.
2. Slides provided by Dr Pirouz during lecture.
3. <https://www.geeksforgeeks.org/binary-heap/>
4. <https://www.techiedelight.com/min-heap-max-heap-implementation-c/>
5. <http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/9-BinTree/heap-insert.html>
6. <https://www.tutorialspoint.com/insertion-into-a-max-heap-in-data-structure>