Isaac Acosta | CS 2302 |

Lab 5 Report

# Introduction

The objective of this lab is to implement Heapsort algorithm. Using percolation techniques to sort the heap array and maintain its heap properties. Followed by the main program, of reading a text file containing integers and apply heapsort algorithm to sort the numbers from least to greatest, or min-heap structure.

# Design and Implementation

The program has two python files, one contains the Heap class and the other one contains the main program that will utilize the Heap class to follow its procedure.

The Heap class has a total of 7 functions.

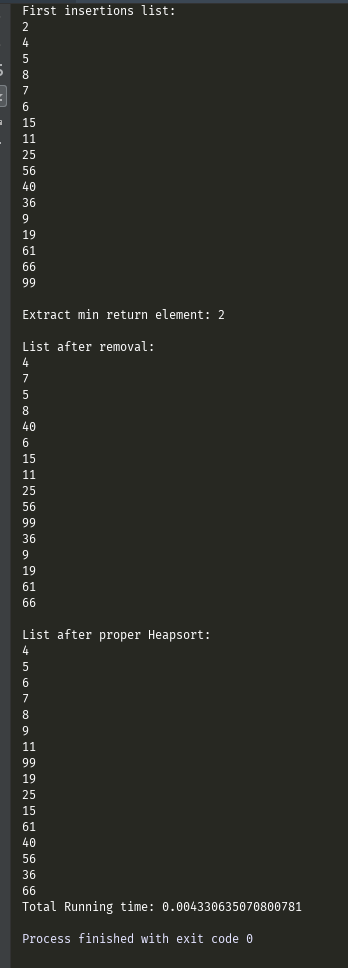
* insert(): Which inserts an element into the heap array. This function percolates up before exiting the function.
* extract\_min(): Extracts the minimal value of the heap array, removes it from the array, and then returns it.
* heap\_sort(): Sorts the elements of the heap array to match the properties of a proper heap array. It percolates down to make a complete heap sort. It sorts by greatest to least, and before returning this list, it is reversed to be from least to greatest.
* percolate\_up(): It adjusts the heap array in a manner that it doesn’t violate any heap rules by starting at some element and going left of the array’s direction, or up in terms of a tree structure.
* percolate\_down(): It adjusts the heap array to fit the rules and properties of a proper heap array. It starts from some element and percolating to the right of the array.
* is\_empty(): It checks if the heap array is empty.
* printer(): It prints the elements of the heap array.

The main program in the second class has two python functions; the def main(), and def read\_file(). The main function of the class starts by calling read\_file(). From here on the program proceeds to read a text file named text.txt (which has a integers separated by commas) and initializing a heap object called min\_heap. Afterwards, line for line the text file is being read. For every line, the elements inside the text file are separated by their commas in between, and therefore inserted into the min\_heap heap object.

The function then proceeds to print three lists into the console. It first prints the elements from min\_heap as they were inserted. It then proceed to call min\_heap.extract\_min(), and then prints the element removed from the heap array along with the heap array in its new order. Lastly, it calls min\_heap.heap\_sort(), and from here on, object min\_heap has a whole different order. It is basically more neatly sorted. The newly sorted heap array is then printed into the console.

# Experimental Results

Screenshot with 17 words:



Conclusion

What was learned from this lab project was how an array can represent a tree structure based on its indexes. It was also learned that min-heaps are the mostly used heap structure in comparison to a max heap. Percolation is important to keep the heap array in maintaining its valid heap properties, whether it be percolating up or down.

**Academic Honesty**

“I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.”