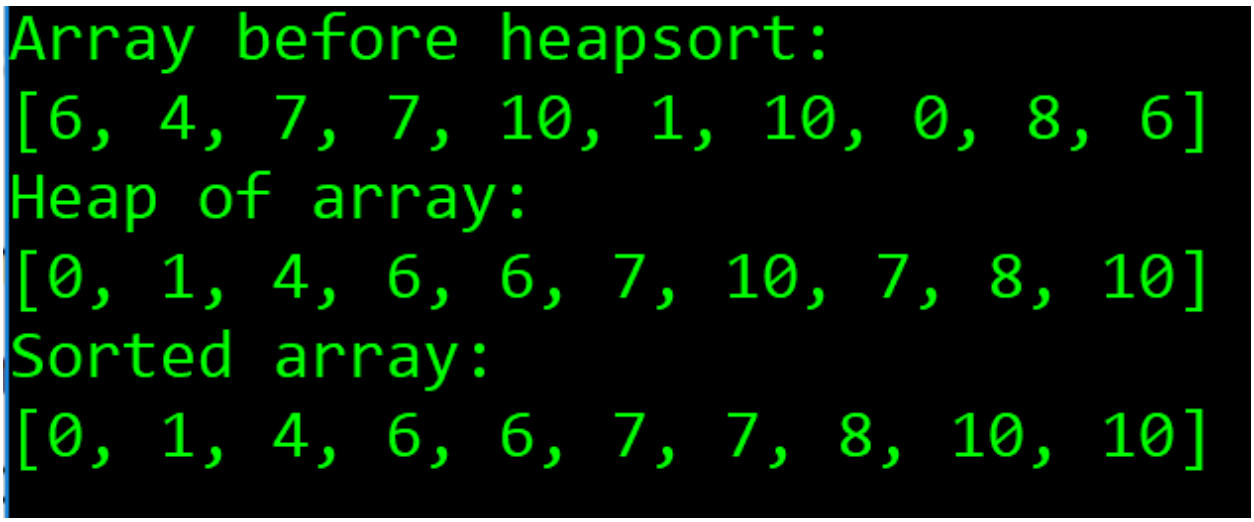
Lab 5
Mark Williams
CS 2302

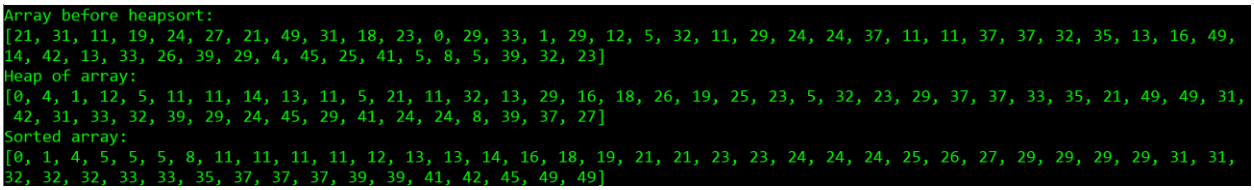
Introduction: In this lab, I create a fully functional heap class. I then use the heap class to implement heap sort on several arrays of integers. I analyze the running time of my approach on these arrays, which are of assorted sizes.

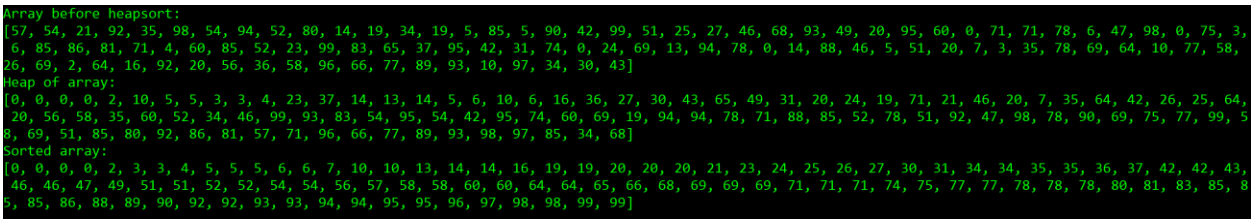
Design Implementation: In this lab, I create a min-heap class. The min-heap class works for all types of variables, not just integers. The class has normal heap functionality, and by that, I mean the ability to insert elements into the heap and retrieve them. To maintain the heap property during insertions and removals, additional functions must be created (percolate up and percolate down) that allow for swapping of elements as insertions or removals occur. The algorithms for all of these functions (insertion, removal, balancing) are standard.

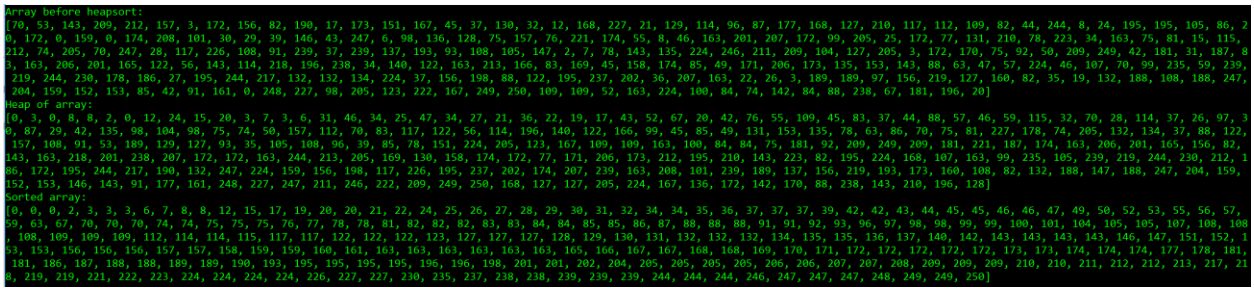
Results: To test the functionality of my heap class, I create arrays of diverse sizes that store random integers. I then transform the array into a heap and remove elements from that heap to get a sorted array.

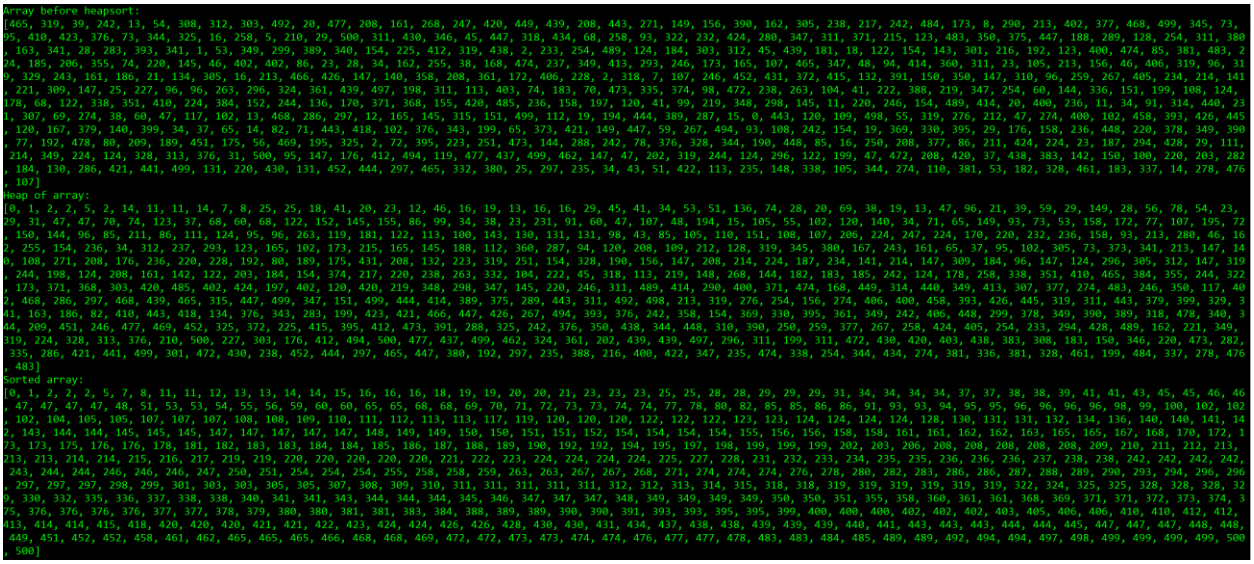
Below are the results of this process for arrays (lists) for arrays of size 10, 50, 100, 250, 500:











The time complexity of running heapsort on the arrays is below:

Array Size	10	50	100	250	500
Time	0.0004966	0.0004964	0.0014873	0.0074637	0.0451415

Conclusion: In this project, I learned how to create a heap class and implement heapsort on an array (list).

Appendix: Code for this lab is located at: https://github.com/mawilliams7/lab5

"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 provide inappropriate assistance to any student in the class."