Gavin Broussard
3102
Programming Assignment 1 explanation

1. Solving the Kary heap

The Kary heap is just like a normal heap except nodes are able to have k children rather than being limited to 2. This allows complexity of O(k*log_k + n) where k is the amount of children that are in each node and n is the max size of the heap. Running time tests was tricky because the time that was spit out to me seemed misleading/inconsistent. I don't know if it is because of the machine I was testing on or because of my code. In short, as the max size of the heap increased, roughly so did the run time. When we incremented the amount of children each node has(k) this decreased our run time. I never implemented a way of printing to an outfile, I just kept the output in the console. This was the case for both the Kary heap and the AVL tree.

2. Solving the AVL tree

This was difficult and I was not able to finish/implement all of the methods I wanted to. I know most of the methods needed could have been implemented using inOrder traversal, I just never figured out how to store and or read the traversal to determine rank and to figure out my select method. I figured out the search method eventually but I am not sure if it is 100% correct/efficient. I know my code for my AVL tree is not correct but tests showed times pretty small leading me to believe everything was kept at O(h) time where h is O(nlogn).

All and all I think this project was a good learning experience. My main programming language has always been python so doing the project in java was tedious. However, I can see how using java, C or C++ would be better than python.