**P5 Testing Document**

Aside from the usual weird Eustis issues, this turned out ok. I had initially given up on trying to make the alphabetizing sort work properly after the frequencies were sorted, but in the 11th hour I figured out how to do it and was very proud. Now to go over the testing files

Wordsort00 is the sample

Wordsort01 is a min case, testing what happens when only one node is created.

Wordsort02 is another mincase, testing what happens if NO nodes are created and a query is attempted. You may think this is redundant but this case actually showed me a core dump error that I’ve now corrected, so this was very valuable.

Wordsort03 is just a medium sized handmade test case that I could eyeball.

Wordsort04 is a large case, created with a program I coded that generates random words and asks for both a type 1 and type 2 action with that word. It’s important to note that for each word, the insertion action comes before the query action, making the query search as exhaustive as possible for testing purposes.

Wordsort05 is the max case. Well, almost max case. I couldn’t actually do a n = 200000 size file in Eustis, as the maximum number of lines was 5012. So it’s n = 5010. Still, I was shocked by how fast it compiled. This used the same program as the previous test file to create its .in contents. This one was valuable because it actually creates duplicate words due to the sheer volume of words, allowing me to verify the sorting is working absolutely perfectly