INF-503 – Homework 5

Problem 1

A)

```
The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW5/./homework
The second argument is: Problem1A

the trie contains 139956 nodes
for 5000 random 36-mers, and all 29868 possible 36-mers:
4593 matches were found with the regular search, and 4593 were found with the fuzzy search

The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW5/./homework
The second argument is: Problem1A

the trie contains 708149 nodes
for 50000 random 36-mers, and all 29868 possible 36-mers:
24255 matches were found with the regular search, and 24255 were found with the fuzzy search

The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW5/./homework
The second argument is: Problem1A

the trie contains 838626 nodes
for 100000 random 36-mers, and all 29868 possible 36-mers:
28855 matches were found with the regular search, and 28855 were found with the fuzzy search
```

As can be see, the size of the trie for 5,000, 50,000, and 100,000 random 36-mers from the COVID genome is 139,956, 708,149 and 838,626 respectively. It makes sense that the number of nodes would be much larger than the number of random sequences, as the maximum possible number of different sequences is 4^n, for sequences of length n, whereas the maximum possible number of nodes is $4^0 + 4^1 + 4^2 + 4^3 + \cdots + 4^n$, for sequences of length n. It also makes sense that the number of nodes is largest, compared to the number of random 36-mers, when the number of random 36-mers is low. When there are many random 36-mers, there's a higher chance that a random 36-mer will share elements with another random 36-mer already on the trie, and thus reduce the number of nodes. Additionally, the number of nodes seem to grow surprisingly little between 50,000 and 100,000 random 36-mers, but this is likely because there are only ~29,900 unique 36-mers to get from the COVID genome, and thus the chances of adding a 36-mer that's not already on the trier are quite small for these amounts of random 36-mers.

The number of matches found for the 5,000, 50,000, and 100,000 random 36-mers from the COVID genome also seem reasonable. For 5,000 random 36-mers, 4593 matches were found. Since we take 5,000 random 36-mers from the COVID genome, and then match every possible 36-mer from the COVID genome with those 5,000 random ones, we should expect to get matches on every random 36-mer. Since the 36-mers are randomly chosen however, it is possible to choose the same 36-mer twice, meaning that only one match count for both of the random 36-mers. This is likely what happened to the missing ~400 random 36-mers that we didn't find matches for here. The same can be said for the 50,000 and 100,000 random runs. Since there are only ~29,900 unique 36-mers to get from the COVID genome, and since the chance of randomly selecting a 36-mer not already on the trie gets lower and lower, the number of matches are hardly growing between 50,000 and 100,000 random 36-mers. We see that for 50,000 random 36-mers, there are still ~5,000 possible unique 36-mers that could be randomly chosen, while after choosing 100,000 random 36-mers, there are only ~100 possible unique 36-mers left.

Lastly, it is apparent that it doesn't matter whether we do fuzzy search or regular search in this case, as the number of matches are the same. We get ~29,900 unique sequences from the COVID genome, but since we are using sequences of length 36, this means that it is possible to get a maximum of 4^{36} unique sequences, and thus it is very unlikely that any two of the ~29,900 sequences will be close enough to each other to only be one mismatch apart.

B)

```
The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW5/./homework
The second argument is: Problem1B

the trie contains 31539 nodes
for 1000 random 36-mers, and all 29868 possible 36-mers:
224 matches were found with the regular search, and 611 were found with the fuzzy search

The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW5/./homework
The second argument is: Problem1B

the trie contains 1237592 nodes
for 50000 random 36-mers, and all 29868 possible 36-mers:
10253 matches were found with the regular search, and 19002 were found with the fuzzy search

The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW5/./homework
The second argument is: Problem1B

the trie contains 2209688 nodes
for 100000 random 36-mers, and all 29868 possible 36-mers:
17012 matches were found with the regular search, and 25962 were found with the fuzzy search
```

I was unsure if the assignment was instructing me to add a 5% chance for error **per 36-mer**, or a 5% chance for error **per character** in all the 36-mers. This implementation have added a 5% chance for error per character.

As can be seen, the number of nodes are now larger than before the random errors were added. This is likely because, by adding random errors, we are also increasing the chances of adding a new unique 36-mer that is not already on the trie, and thus we increase the chance that new nodes will be added.

As can be seen, the number of matches are now quite low for the regular search, but still reasonable for the fuzzy search. This is likely because the regular search can't handle the random errors. Each character in the 36-mer have a 5% chance to change into another character (actually the chance is lower than 5%, as the character has a 25% chance to change into the same character as it was before), and thus, the regular search are likely only finding all the 36-mers that didn't happen to generate errors. The fuzzy search on the other hand should be able to find all the 36-mers that either didn't have any errors, or that have an error on only a single character. Thus, the fuzzy search is performing almost as good as if there weren't any errors, although it likely isn't picking up the 36-mers with two or more errors.