INF-503 – Homework 6

Problem 1

1)

```
The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem1A
The third argument is: 5000

the trie contains 446931004 nodes
Found fragment nr 1 from appendix B
Found fragment nr 2 from appendix B
Did not find fragment nr 3 from appendix B
Did not find fragment nr 4 from appendix B
Found fragment nr 5 from appendix B
```

As can be seen, the suffix trie and the perfect match search was implemented correctly. The search function found fragments 1, 2, and 5 from Appendix B in the COVID genome.

2)

```
The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem1B
The third argument is: 5000

the trie contains 446931004 nodes
It took 33 seconds to build the suffix trie
It took 0 seconds to search for all 5000 random sequences
Out of 5000 random sequences from the genome, 5000 were found


The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem1B
The third argument is: 50000

the trie contains 446931004 nodes
It took 32 seconds to build the suffix trie
It took 0 seconds to search for all 50000 random sequences
Out of 50000 random sequences from the genome, 50000 were found


The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem1B
The third argument is: 100000

the trie contains 446931004 nodes
It took 33 seconds to build the suffix trie
It took 0 seconds to search for all 100000 random sequences
Out of 100000 random sequences from the genome, 100000 were found
```

As can be seen from the output, the suffix trie containing the COVID genome has a total of 446,931,004 nodes. The search function consistently finds every randomly generated sequence, which makes sense, since the randomly generated sequences are generated from the COVID genome itself.

The suffix trie takes a significant amount of time to construct, at around 33 seconds.

For 5,000, 50,000 and 100,000 randomly generated sequences, it took 0 seconds to complete the search, making it hard to estimate the big O notation. I therefore tried the search again, this time using 1, 5, 10, 20, and 50 million randomly generated sequences. The results are shown in the table below:

Nr of random sequences	1,000,000	5,000,000	10,000,000	20,000,000	50,000,000
Time (s)	1	7	14	27	69

Judging from the results, the time to search n randomly generated sequences seems to scale in a linear fashion, and thus the big O notation should be O(n)

Problem 2 (extra credit)

1)

```
The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem2A
The third argument is: 5000

the trie contains 49003 nodes
Found fragment nr 1 from appendix B
Found fragment nr 2 from appendix B
Did not find fragment nr 3 from appendix B
Did not find fragment nr 4 from appendix B
Found fragment nr 5 from appendix B
```

As can be seen, the suffix tree and search function was implemented correctly. The search function found fragments nr 1, 2, and 5 from Appendix B, the same fragments that were found when using the suffix trie.

2)

```
The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem2B
The third argument is: 5000

the trie contains 49003 nodes
It took 0 seconds to build the suffix tree
It took 0 seconds to search for all 5000 random sequences
Out of 5000 random sequences from the genome, 5000 were found


The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem2B
The third argument is: 50000

the trie contains 49003 nodes
It took 0 seconds to build the suffix tree
It took 0 seconds to search for all 50000 random sequences
Out of 50000 random sequences from the genome, 50000 were found


The number of arguments passed: 3
The first argument is: /home/fvg6/ondemand/LS_Datastructures_HW6/./homework
The second argument is: Problem2B
The third argument is: 100000

the trie contains 49003 nodes
It took 0 seconds to build the suffix tree
It took 0 seconds to search for all 100000 random sequences
Out of 100000 random sequences from the genome, 100000 were found
```

As can be seen from the output, the suffix tree containing the COVID genome has a total of 49,003 nodes, much fewer than the suffix trie had. This is likely due to the suffix tree using complex nodes, instead of just having one node per character. The search function still finds every randomly generated sequence, which again makes sense, since the randomly generated sequences are generated from the COVID genome itself.

The suffix tree takes much less time to construct than the suffix trie, which makes sense, since the big O notation for construction of a suffix trie is $O(n^2)$, for a sequence of length n, while the big O notation for construction of a suffix tree is only $O(n)$.

I again tested the search function, using 1, 5, 10, 20, and 50 million randomly generated sequences. The results are shown in the table below:

Nr of random sequences	1,000,000	5,000,000	10,000,000	20,000,000	50,000,000
Time (s)	0	4	8	15	36

From the results, we see that the time to search n randomly generated sequences is lower for the suffix tree than for the suffix trie. The time still seems to scale in a linear fashion however, and thus the big O notation should be O(n).