**Homework #5**

**PART - A**

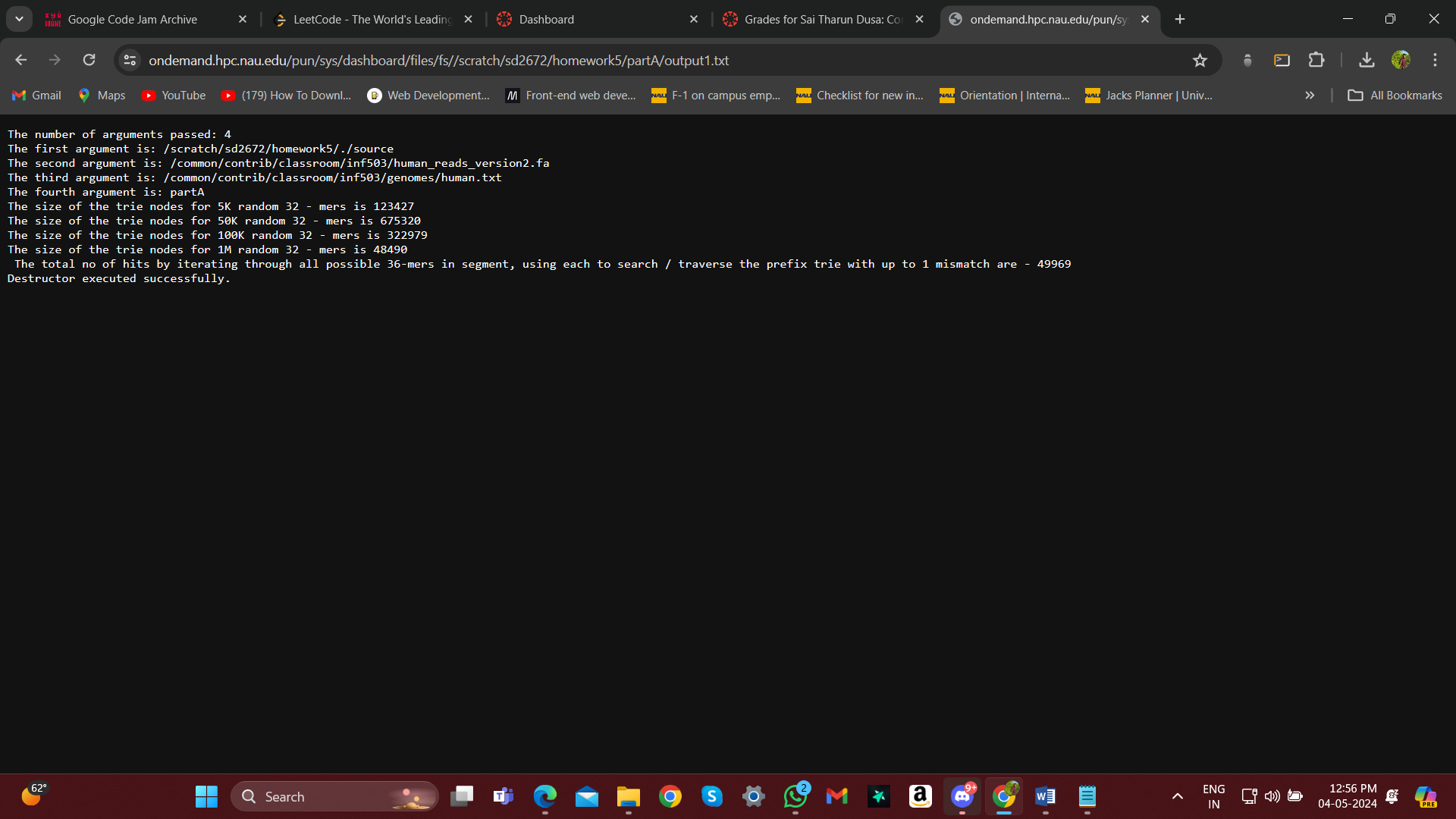
Basic prefix trie: Pick a random 50K long segment from the human genome assembly. Generate 5K,

50K, 100K, and 1M random 32-mers this segment and store them in the prefix trie. Hint: generate a

random starting position somewhere in the segment and read 32 characters starting from that

position.

* **For each of the 32-mer datasets, what are the sizes of the trie (# of nodes)? Explain the pattern that you observed.**



5K: 123427

50K: 675320

100K:322979

1M:48490

The length of the segment where we are taking the 32 – mers is 50K. So definitely, the size will increase for 5k iterations to 50k iterations. And later, size for 100K and 1M iterations decreases because of the duplicate 32-mers. Because, nodes for the characters of those 32 – mers would already exists in the trie data structure. So we can conclude that the size of the trie increases till the length of the given segment iterations, in our case it is 50K. And then it decreases because of duplicates.

• **Iterate through all possible 32-mers in the segment, using each to search / traverse the prefix trie with up to 1 mismatch. How many of your 32-mers had a match? Does it make sense? Explain why.**

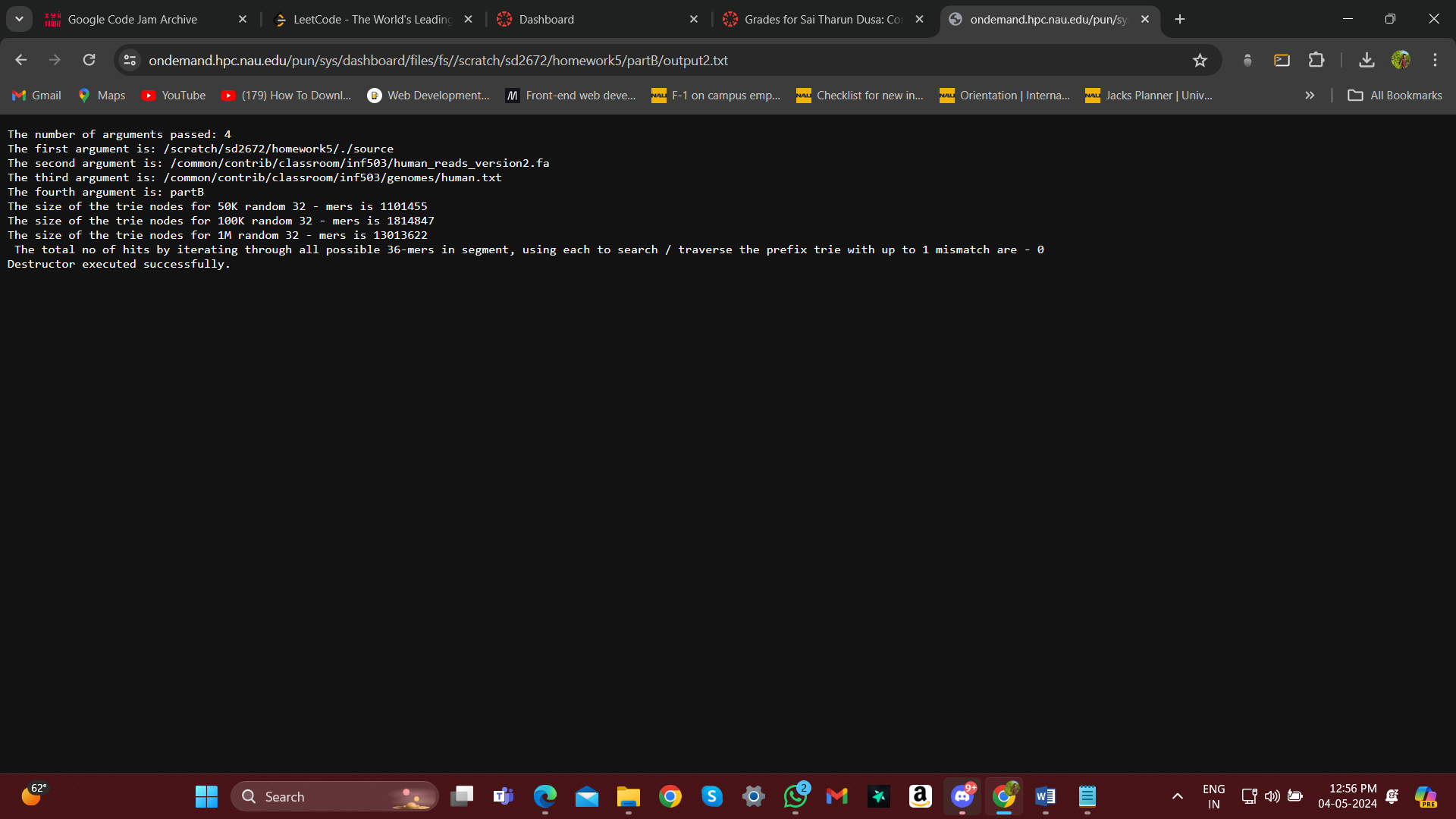
There are total 49969 hits. And it does makes sense, because we have done 5K,50K,100K,1M iterations over this 50K segment. So after all the iterations, definitely all the 32- mers of the 50Ksegment would be in the trie data structure.

All 32-mers of 50K segment would be 50000 – 32 +1 which is 49969 and it is equal to the total no of hits.

**PART – B**

(20pts) Impact of error rate on trie structure: Use the same random 50K long segment from the human genome assembly that you used in part A. Generate 5K, 50K, 100K, and 1M random 36-mers from this segment with 5% per-base error rate and store them in the prefix trie. Hint: repeat the process from part A, except each base of 36-mer has a 5% chance of mutation/error.

* **For each of the 32-mer datasets, what are the sizes of the trie (# of nodes)? Explain differences (if any) between the trie sizes in partA and part B.**



50K: 1101455

100K:1814847

1M:13013622

The trie sizes for part A would increase upto some point and then it decreases because of the

duplicates. But for part B, we have inserted the 32 – mers with 5% chance of mutation/error, so

there is no chance of duplicates. A new path with new nodes for every new 32-mer are created in

the trie data structure. So the size never decreases in partB, the size just increases.

* **Iterate through all possible 32-mers in segment, using each to search / traverse the prefix trie with**

**up to 1 mismatch. How many of your 32-mers had a match? Does it make sense? Explain why.**

Total hits – 0.

For every 32 – mer from the 50K segment, a 5 % chance of error is maintained and then it is inserted

into the trie data structure. And later when we do search for 32 – mers of 50K segment without

maintaining any errors, then definitely, it is impossible to find the match with upto 1mismatch.

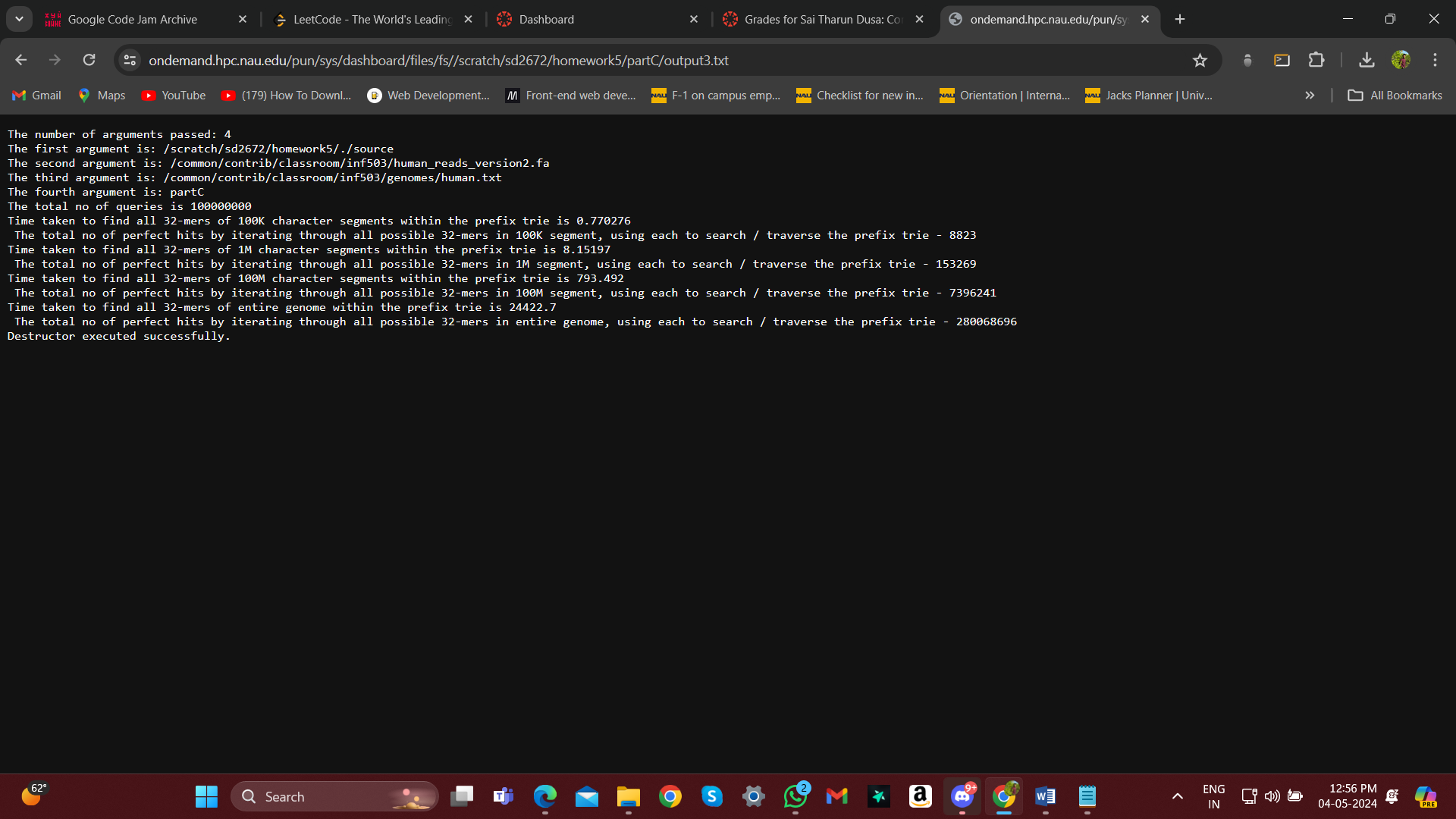
Because, there are no paths for the actual 32 – mers in the trie data structure. Since, we generate

the 5% error randomly, there might be a chance of getting hits upto 2 or 3.

**PART – C**

(20pts) Full prefix trie experience: Load the entire query dataset into the prefix trie. Generate a random segment of the human genome of size 100K, 1M, and 100M characters and use the 32-mers in this this segment to search against the prefix trie. For this problem, only look for perfect matches (no need for fuzzy matching).

* **How long did it take you to find all 32-mers of 100K, 1M, and 100M character segments within the prefix trie? Estimate how long it would take to search the entire human genome.**



**NOTE : The time taken in the above output is in seconds.**

Time taken for 32 – mers of:

100K segement – 0.77 seconds.

1M segment – 8.15 seconds.

100M segment – 793.4 seconds.

For entire genome – 24422.7 seconds.

* **How many ‘hits’ did you find for 100K, 1M, and 100M segments? Estimate how many you would find in the full genome.**

Total no of hits for:

100K segment – 8823

1M segment – 1523269

100M segment – 7396241

For entire genome - 280068696

**Steps of execution:**

* Created total of three files main.cpp, header\_definitions.cpp and header.h
* header.h file contains all the header files that are used in the program
* The main.cpp contains the main function and all the function calls required to get desired output
* header\_definitions.cpp file contains all the function definitions which are declared in the header file
* Created a make file to run the code
* Uploaded all the above files to a directory on monsoon
* There I have opened terminal and entered the command “make” then source executable file is generated next we need to run the source file with file path and the part of execution.
* The command for execution is

./source /common/contrib/classroom/inf503/human\_reads\_version2.fa /common/contrib/classroom/inf503/genomes/human.txt partA

./source /common/contrib/classroom/inf503/human\_reads\_version2.fa /common/contrib/classroom/inf503/genomes/human.txt partB

./source /common/contrib/classroom/inf503/human\_reads\_version2.fa /common/contrib/classroom/inf503/genomes/human.txt partC