**Homework #5**

**Km3247@nau.edu**

Each homework submission must include:

* An archive (.zip or .gz) file of the source code containing:

o The makefile used to compile the code on Monsoon **(5pts)** o All .cpp and .h files **(5pts)**

* A full write-up (.pdf of .doc) file containing answers to homework’s questions **(5pts)**, including the exact command line needed to execute every subproblem of the homework

The source code must follow the following guidelines:

* No external libraries that implement data structures discussed in class are allowed, unless specifically stated as part of the problem definition. Standard input/output and utilities libraries (e.g. math.h) are ok.
* All external data sources (e.g. input data) must be passed in as a command line argument (no hardcoded paths within the source code **(5pts)**.
* Solutions to sub-problems must be executable separately from each other. For example, via a special flag passed as command line argument **(5pts)**

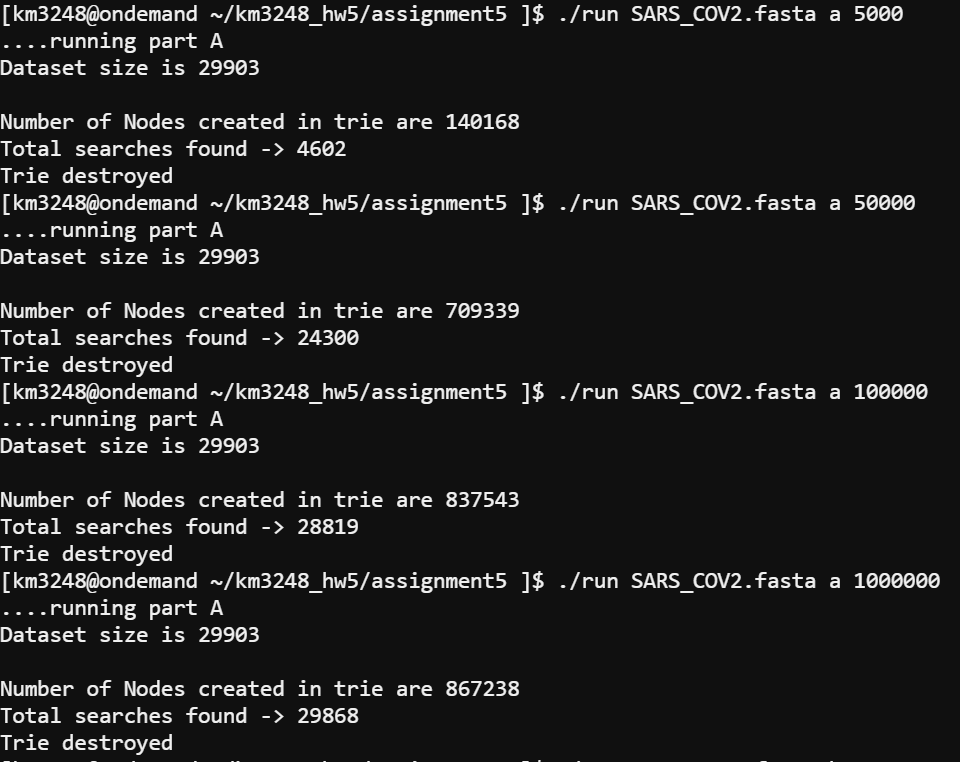
# Problem #1 (of 1): Prefix Trie

Create a class called ***prefix\_trie***. The purpose of the class will be to contain a FASTA read set and all of the functions needed to operate on this set. Use the **prefix trie** data-structure to store the genomic sequences of the given read dataset. Here you will be performing fuzzy matching, tolerating up to 1 mismatch.

At minimum, the class must contain(**25pts**):

* A default constructor
* At least one custom constructor to build a trie from a set of queries (of size n)
* A function to traverse (*search*) the trie using a genome of size G. Note that you can assume that G >> n. You will need to implement a ***fuzzy search tolerating up to 1 mismatch* (substitutions only)**. Hint: use a stack to keep track of branches in the tree that need to be explored.
* A destructor
* A copy constructor

1. (**25pts) Basic prefix trie**: Generate 5K, 50K, 100K, and 1M random 36-mers from the SARS-CoV2 genome sequence (Appendix A) and store them in the prefix trie. Hint: generate a random starting position somewhere in the genome and read 36 characters starting from that position.



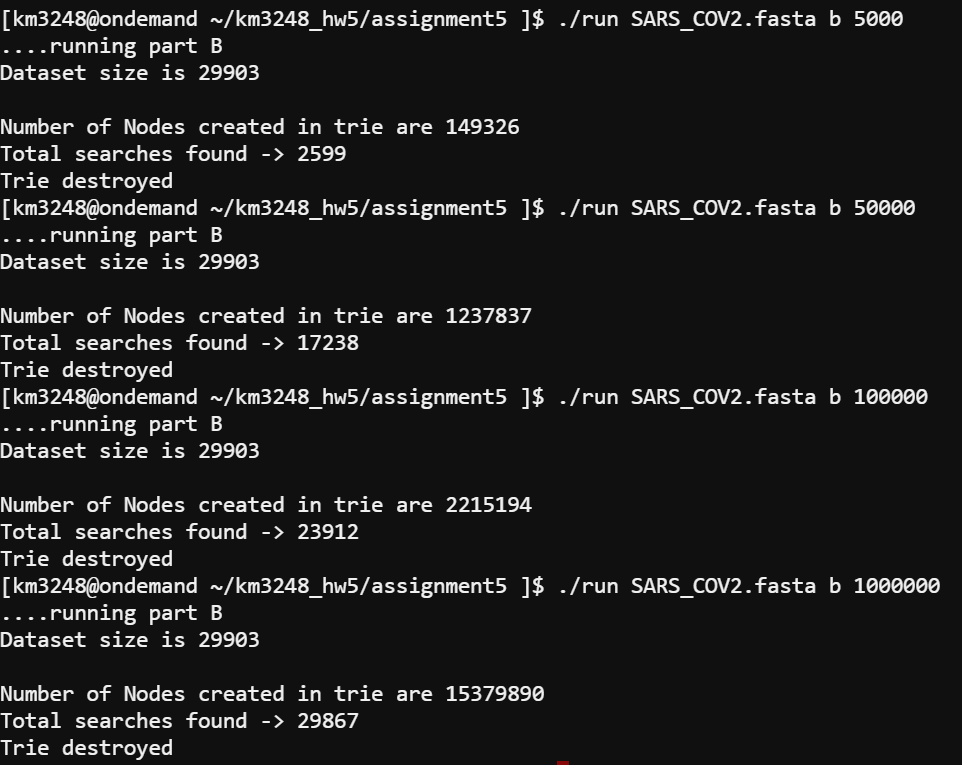
* + For each of the 36-mer datasets, what are the sizes of the trie (# of nodes)? Explain the pattern that you observed.

Number of nodes will increment in a quick way when the trie is vacant, until a particular point. The quantity of nodes will increment gradually as more 36mers are embedded. This example is noticed on the grounds that the trie doesn't create a nodes when already existing.

* + Iterate through all possible 36-mers in the SARS-CoV2 genome, using each to search / traverse the prefix trie with up to 1 mismatch. How many of your 36-mers had a match? Does it make sense? Explain why.

Number of matches for n inquiries id not exactly or equivalent to the quantity of embedded groupings up to a greatest breaking point. Say for instance, for 5000 additions, the matches can be 5000 assuming all embedded arrangements are different to one another. Since the inclusions are arbitrary, there is chance that similar additions are rehashed in those 5000 groupings, so number of matches will be under 5000. As the quantity of additions expands, the trie contains all the succession of the trie, and adding additional inclusions from a similar genome won't add additional hubs. So you number of matches when embedded 1M 36mrrs will be equivalent to by embedding 2M, etc

1. **(25pts) Impact of error rate on trie structure:** Generate 5K, 50K, 100K, and 1M random 36-mers from the SARS-CoV2 genome sequence 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 36-mer datasets, what are the sizes of the trie (# of nodes)? Explain differences (if any) between the trie sizes in partA and part B.

Number of nodes will increment in a quick way, creating more than in part A for same size as there is error decreasing chance of repeatition in nodes.

* + Iterate through all possible 36-mers in the SARS-CoV2 genome, using each to search / traverse the prefix trie with up to 1 mismatch. How many of your 36-mers had a match?

Does it make sense? Explain why.

Due to mutation of inserted 36mers with 5% error, some queries inserted will not match resulting in lesser number of matches than that observed in part A