HOMEWORK#2

km3248@nau.edu

A. Read in the entire ~36 million read set (query read set) and store it in the FASTAreadset_LL class. Implement a search function which would take a sequence fragment (OK to assume that it will be exactly 50 characters long) and search for this fragment within the FASTAreadset_LL object. The search function should return the pointer to the node containing the match OR the NULL pointer value if a 'hit' was not found.

• Which of the following sequences were found in the read set?

Command: ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa A

```
[km3248@ondemand ~/assignment2 ]$ ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa A
0
Reading 36 Million fragments

fragnments in linked list: 36220411

CTAGGTACATCCACACACAGCAGCGCATTATGTATTTATTGGATTTATTT Found at position 82 in fasta dataset

GCGCGATCAGCTTCGCGCGCACCGCGAGCGCCGATTGCACGAAATGGCGCFound at position 36220407 in fasta dataset

CGATGATCAGGGGCGTTGCGTAATAGAAACTGCGAAGCCGCTCTATCGCC Not found in fasta dataset

CGTTGGGAGTGCTTGGTTTAGCGCAAATGAGTTTTCGAGGCTATCAAAAAFound at position 1525 in fasta dataset

ACTGTAGAAGAAAAAAGTGAGGCTGCTCTTTTACAAGAAAAAGTNNNNNNFound at position 1524 in fasta dataset
 deallocation started
Successfully de-allocated the memory used for storing linkedlist
```

• Would sorting the linked list help speed up the search (on average and in the worst case)? Explain why or why not?

If linked list is already sorted then it will increase search speed but using Sorting algorithm for searching fragments for this problem is might not help in speed up the search because first, we are searching only 5 sequences in 36 million and for sorting the average time complicity and worst time complexity is O(nlogn) and for searching it is O(logn). In general, if we have 5 million sequence fragments to be searched in 36 million fragments then sorting these 36 million Will definitely speed up the search.

B. Read in the Bacillus anthracis genome into a character array (you will need to determine the exact size of the sequence). Iterate through all possible 50-character long fragments within the genome by shifting fragment start location by one character each time. Use these fragments to search within the FASTAreadset_LL object.

• How many 50 character fragments can you make from the B. anthracis genome?

Command: ./homework2 /common/contrib/classroom/inf503/test_genome.fasta 1B

```
[km3248@ondemand ~/assignment2 ]$ ./homework2 /common/contrib/classroom/inf503/test_genome.fasta 1B
0
reading genome dataset
Total 5227244 50charcter sequences found at genome dataset
 deallocation started
Successfully de-allocated the memory used for storing linkedlist
```

• What is the overlap between the genome's 50-mers and the ~36 million fragments you've stored in the FASTAreadset_LL object? Please note that depending on the efficiency of your algorithm, this step may take a long time. First estimate the total time using 1,000, 10,000, and 100,000 queries – if total time estimate is greater than 24 CPU hours, provide estimate rather than exact number.

First 1000 50-mers:

Command: srun --mem=6000 --time=10:00:00 ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa 2B /common/contrib/classroom/inf503/test_genome.fasta 1000

```
[km3248@ondemand ~/assignment2 ]$ ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa 2B /common/contrib/classroom/inf503/test_genome.fasta 1000
1000
Reading 36 million fragments

fragnments in linked list: 36220411
Reading Genome data
The total 1000 50mers matched in 36 million is 798

time taken for problem2B with 1000 50mers is 439.08 seconds
 deallocation started
Successfully de-allocated the memory used for storing linkedlist
```

Time taken to run search for first 1000 50-mers is 439 seconds(7.3 minutes)

First 10000 50-mers:

Command: srun --mem=6000 --time=10:00:00 ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa 2B /common/contrib/classroom/inf503/test_genome.fasta 10000

```
[km3248@ondemand ~/assignment2 ]$ srun --mem=60000 ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa 2B /common/contrib/classroom/inf503/test_ge
nome.fasta 10000
srun: job 50450784 queued and waiting for resources
srun: job 50450784 has been allocated resources
10000
Reading 36 million fragments

fragnments in linked list: 36220411
Reading Genome data
The total 10000 50mers matched in 36 million is 8373

time taken for problem2B with 10000 50mers is 2120.18 seconds
 deallocation started
Successfully de-allocated the memory used for storing linkedlist
```

Time taken to search for first 10000 50-mers is 2120(35.3 minutes)

First 50000 50-mers:

Command: srun --mem=6000 --time=10:00:00 ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa 2B /common/contrib/classroom/inf503/test_genome.fasta 50000

```
[km3248@ondemand ~/assignment2 ]$ srun --mem=60000 --time=10:00:00 ./homework2 /common/contrib/classroom/inf503/hw_dataset.fa 2B /common/contrib/classroo
m/inf503/test_genome.fasta 50000
srun: job 50453094 queued and waiting for resources
srun: job 50453094 has been allocated resources
50000
Reading 36 million fragments

fragnments in linked list: 36220411
Reading Genome data
The total 50000 50mers matched in 36 million is 42199

time taken for problem2B with 50000 50mers is 11110.12 seconds
 deallocation started
Successfully de-allocated the memory used for storing linkedlist
```

Time taken to search for first 50000 50-mers is 11110 seconds(3hrs)

For 5.2 million 50-mers it would take approximately 300hrs to search.

• You've iterated through all 50-mers found in the genome and used them to search within the query read set. Would it have been faster to flip the problem – i.e. store the genome's fragments in a data structure and iterate through the query read set? Explain why or why not.

Yes if we use hash tables it would have be less time. I don't think that flipping the problem is a good idea because now we are searching n-5.2 million in m-36 million then the TC is O(n)*O(logm). If we filp the problem then the TC will be O(m)*O(logn) which takes more time than the current problem.