INF503_Assignment_4_mp2525 - Solutions

Problem #1 (of 2): Implement Smith-Waterman algorithm

A. Write a function in C++ that would implement the Smith-Waterman alignment between two genomic sequences.

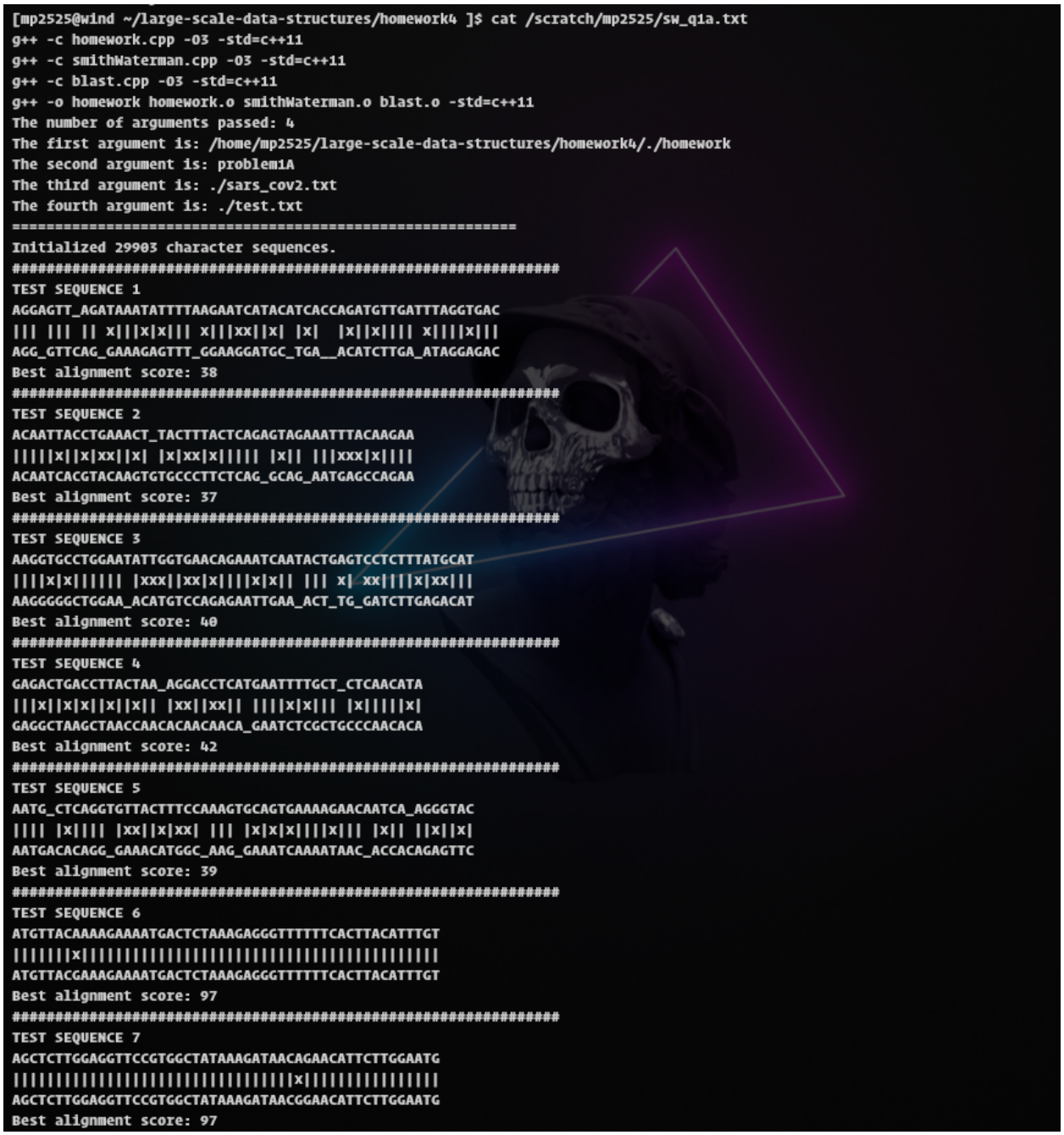
- The function must take a genomic similarity scoring matrix (+2 for match, -1 for mismatch) and a gap penalty (-3) as an input.
- The function must return:
    - score for best alignment
    - alignment in text format (hint: use a struct of 3 character arrays, 2 for sequences one for alignment codes {x, I, } – use whitespace for gaps). See below.
- Test your code using the SARS-COV2 viral genome found in appendix A at the bottom of this homework assignment and sequence fragments found in appendix B at the bottom of this homework.
- You will need to submit a screenshot of the output as part of the homework write up (it should look something like the text below)

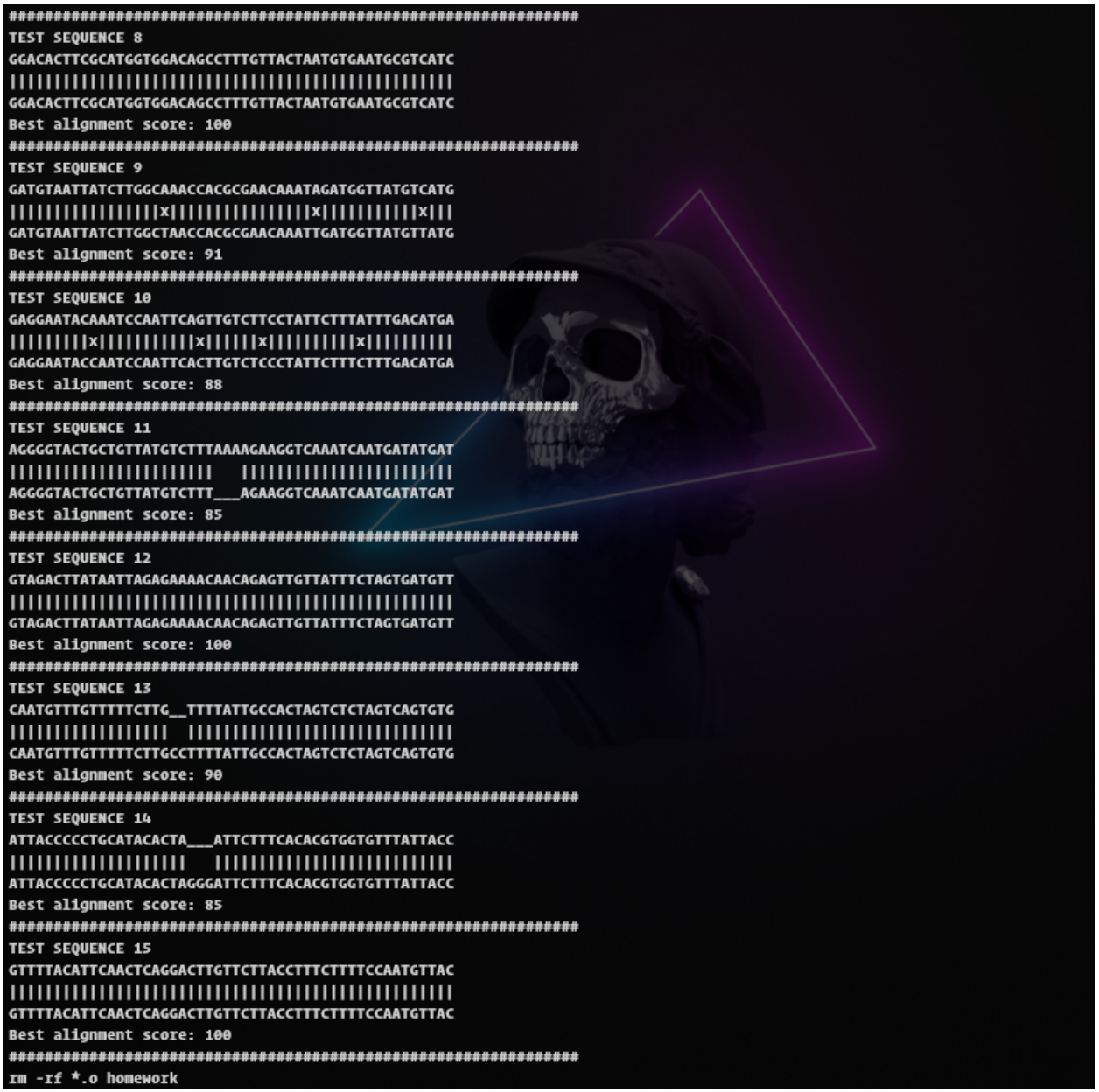
```
GATTA
| x||
G_CTA
```

Ans:

I have created a smithWaterman function that takes two sequences, gap penalty, match, mismatch, and doPrint. When doPrint is set to true, genome sequence alignments are printed. The function returns the best alignment score.

After testing the code using the SARS-COV2 viral genome found in appendix A at the bottom of this homework assignment and sequence fragments found in appendix B at the bottom of this homework, we get the result as shown in the screenshot below.


[mp2525@wind ~/large-scale-data-structures/homework4]$ cat /scratch/mp2525/sw_q1a.txt
g++ -c homework.cpp -O3 -std=c++11
g++ -c smithWaterman.cpp -O3 -std=c++11
g++ -c blast.cpp -O3 -std=c++11
g++ -o homework homework.o smithWaterman.o blast.o -std=c++11
The number of arguments passed: 4
The first argument is: /home/mp2525/large-scale-data-structures/homework4/./homework
The second argument is: problem1A
The third argument is: ./sars_cov2.txt
The fourth argument is: ./test.txt
==========
Initialized 29903 character sequences.
##########
TEST SEQUENCE 1
AGGAGTT_AGATAAATATTTTAAGAATCATACATCACCAGATGTTGATTTAGGTGAC
||| ||| || x|||x|x||| x|||xx||x| |x| |x||x|||| x||||x|||
AGG_GTTCAG_GAAAGAGTTT_GGAAGGATGC_TGA__ACATCTTGA_ATAGGAGAC
Best alignment score: 38
##########
TEST SEQUENCE 2
ACAATTACCTGAAACT_TACTTTACTCAGAGTAGAAATTTACAAGAA
|||||x||x|xx||x| |x|xx|x|||| |x|| |||xxx|x||||
ACAATCACGTACAAGTGTGCCCTTCTCAG_GCAG_AATGAGCCAGAA
Best alignment score: 37
##########
TEST SEQUENCE 3
AAGGTGCCTGGAATATTGGTGAACAGAAATCAATACTGAGTCCTCTTTATGCAT
||||x|x|||||| |xxx||xx|x||||x|x|| ||| x| xx||||x|xx|||
AAGGGGGCTGGAA_ACATGTCCAGAGAATTGAA_ACT_TG_GATCTTGAGACAT
Best alignment score: 40
##########
TEST SEQUENCE 4
GAGACTGACCTTACTAA_AGGACCTCATGAATTTTGCT_CTCAACATA
|||x||x|x||x||x|| |xx||xx|| ||||x|x||| |x|||||x|
GAGGCTAAGCTAACCAACACAACAACA_GAATCTCGCTGCCCAACACA
Best alignment score: 42
##########
TEST SEQUENCE 5
AATG_CTCAGGTGTTACTTTCCAAAGTGCAGTGAAAAGAACAATCA_AGGGTAC
|||| |x|||| |xx||x|xx| ||| |x|x|x|||x||| |x|| ||x||x|
AATGACACAGG_GAAACATGGC_AAG_GAAATCAAAATAAC_ACCACAGAGTTC
Best alignment score: 39
##########
TEST SEQUENCE 6
ATGTTACAAAAGAAAATGACTCTAAAGAGGGTTTTTTCACTTACATTTGT
|||||||x|||||||||||||||||||||||||||||||||||||||||||
ATGTTACGAAAGAAAATGACTCTAAAGAGGGTTTTTTCACTTACATTTGT
Best alignment score: 97
##########
TEST SEQUENCE 7
AGCTCTTGGAGGTTCCGTGGCTATAAAGATAACAGAACATTCTTGGAATG
|||||||||||||||||||||||||||||||x||||||||||||||||||
AGCTCTTGGAGGTTCCGTGGCTATAAAGATAACGGAACATTCTTGGAATG
Best alignment score: 97


##
TEST SEQUENCE 8
GGACACTTCGCATGGTGGACAGCCTTTGTTACTAATGTGAATGCGTCATC
|||||||||||||||||||||||||||||||||||||||||||||||||||
GGACACTTCGCATGGTGGACAGCCTTTGTTACTAATGTGAATGCGTCATC
Best alignment score: 100
##
TEST SEQUENCE 9
GATGTAATTATCTTGGCAAACCACGCGAACAAATAGATGGTTATGTCATG
||||||||||||||||x|||||||||||||||x||||||||||x|||
GATGTAATTATCTTGGCTAACCACGCGAACAAATTGATGGTTATGTTATG
Best alignment score: 91
##
TEST SEQUENCE 10
GAGGAATACAAATCCAATTCAGTTGTCTTCCTATTCTTTATTTGACATGA
|||||||||x||||||||||x|||||x|||||||||||x||||||||||
GAGGAATACCAATCCAATTCACTTGTCTCCCTATTCTTTCTTTGACATGA
Best alignment score: 88
##
TEST SEQUENCE 11
AGGGGTACTGCTGTTATGTCTTTAAAAGAAGGTCAAATCAATGATATGAT
||||||||||||||||||||||||| |||||||||||||||||||||||||
AGGGGTACTGCTGTTATGTCTTT___AGAAGGTCAAATCAATGATATGAT
Best alignment score: 85
##
TEST SEQUENCE 12
GTAGACTTATAATTAGAGAAAACAACAGAGTTGTTATTTCTAGTGATGTT
|||||||||||||||||||||||||||||||||||||||||||||||||||
GTAGACTTATAATTAGAGAAAACAACAGAGTTGTTATTTCTAGTGATGTT
Best alignment score: 100
##
TEST SEQUENCE 13
CAATGTTTGTTTTTCTTG__TTTTATTGCCACTAGTCTCTAGTCAGTGTG
||||||||||||||||| |||||||||||||||||||||||||||||||
CAATGTTTGTTTTTCTTGCCTTTTATTGCCACTAGTCTCTAGTCAGTGTG
Best alignment score: 90
##
TEST SEQUENCE 14
ATTACCCCCTGCATACACTA___ATTCTTTCACACGTGGTGTTTATTACC
||||||||||||||||||| |||||||||||||||||||||||||||||
ATTACCCCCTGCATACACTAGGGATTCTTTCACACGTGGTGTTTATTACC
Best alignment score: 85
##
TEST SEQUENCE 15
GTTTTACATTCAACTCAGGACTTGTTCTTACCTTTCTTTTCCAATGTTAC
|||||||||||||||||||||||||||||||||||||||||||||||||||
GTTTTACATTCAACTCAGGACTTGTTCTTACCTTTCTTTTCCAATGTTAC
Best alignment score: 100
##
rm -rf *.o homework

B. Generate 1K, 10K, 100K, and 1M (million) completely random genomic sequences (50nt) to use as targets for alignment and use SARS-COV2 genome as the subject. Perform alignment of the queries to the subject sequence and record time to completion (in seconds/minutes).

Ans:

```
[mp2525@wind ~ ]$ cat /scratch/mp2525/sw_q1b.txt
g++ -c homework.cpp -O3 -std=c++11
g++ -o homework homework.o smithWaterman.o blast.o -std=c++11
The number of arguments passed: 3
The first argument is: /home/mp2525/large-scale-data-structures/homework4/./homework
The second argument is: problem1B
The third argument is: ./sars_cov2.txt
=======================================================
Initialized 29903 character sequences.
Time to test Smith Waterman with 1000 sequence: 20.154 seconds.
Time to test Smith Waterman with 10000 sequence: 200.248 seconds.
rm -rf *.o homework
```

I have generated 1k and 10k completely random sequences to use as targets for alignments and used SARS-COV2 genome as the subject. I give up on 100k and 1M random sequences because of the huge execution time. I haven't printed the sequence alignments for this question, the function parameter doPrint can be set to true in smithWaterman function to view the sequence alignment.

Task	Random Sequences	Execution time
1	1,000	20.154 seconds
2	10,000	200.248 seconds
3	100,000	2,000 seconds (estimated)
4	1,000,000	20,000 seconds (estimated)

Problem #2 (of 2): Having a BLAST

A. Implement a seed-based Smith-Waterman. This means:

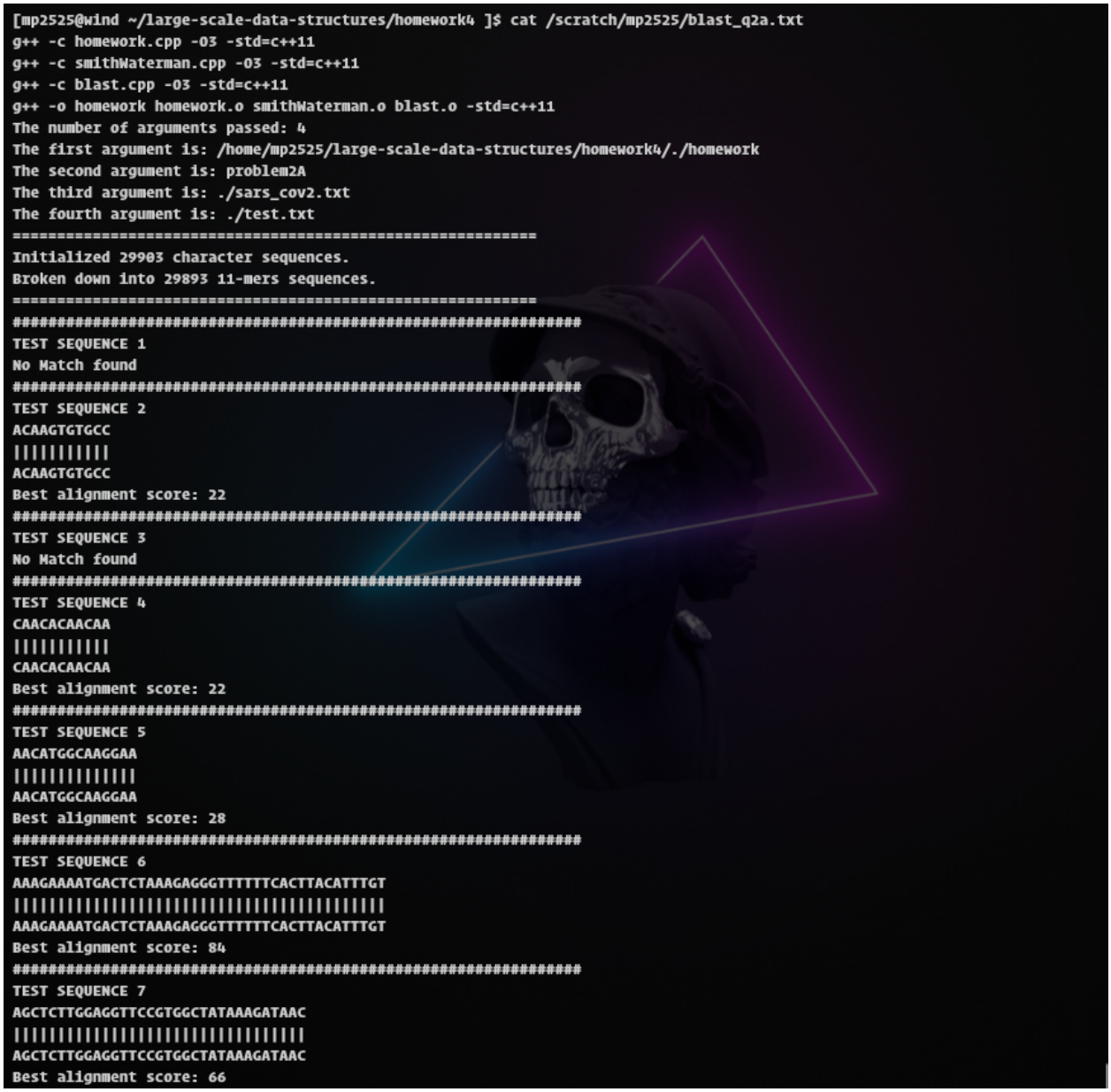
- Use the genome in Appendix A and break I down into seeds with word size = 11.
- Load your seeds (created in part A) into memory
- For each read disassemble it into k-mers (of size 11). Compare your read k-mers to the SARS-COV2 seeds.
- If a seed match is found, extend the seed by cutting out the appropriate segment of the subject (Genome of SARS-COV2) and running the Smith-Waterman on the two sequences (original read and the segment from SARS-COV2)
    - Beware of edge cases
    - Ok to just expand one seed and be done (multiple seeds from a read can be found, typically necessitating multiple seed expansions and decision on what is the best alignment)
- Test your code on the 50-mers I have provided below in Appendix B. You must report alignment for the 50-mers I've provided as part of the homework solution.

Ans:

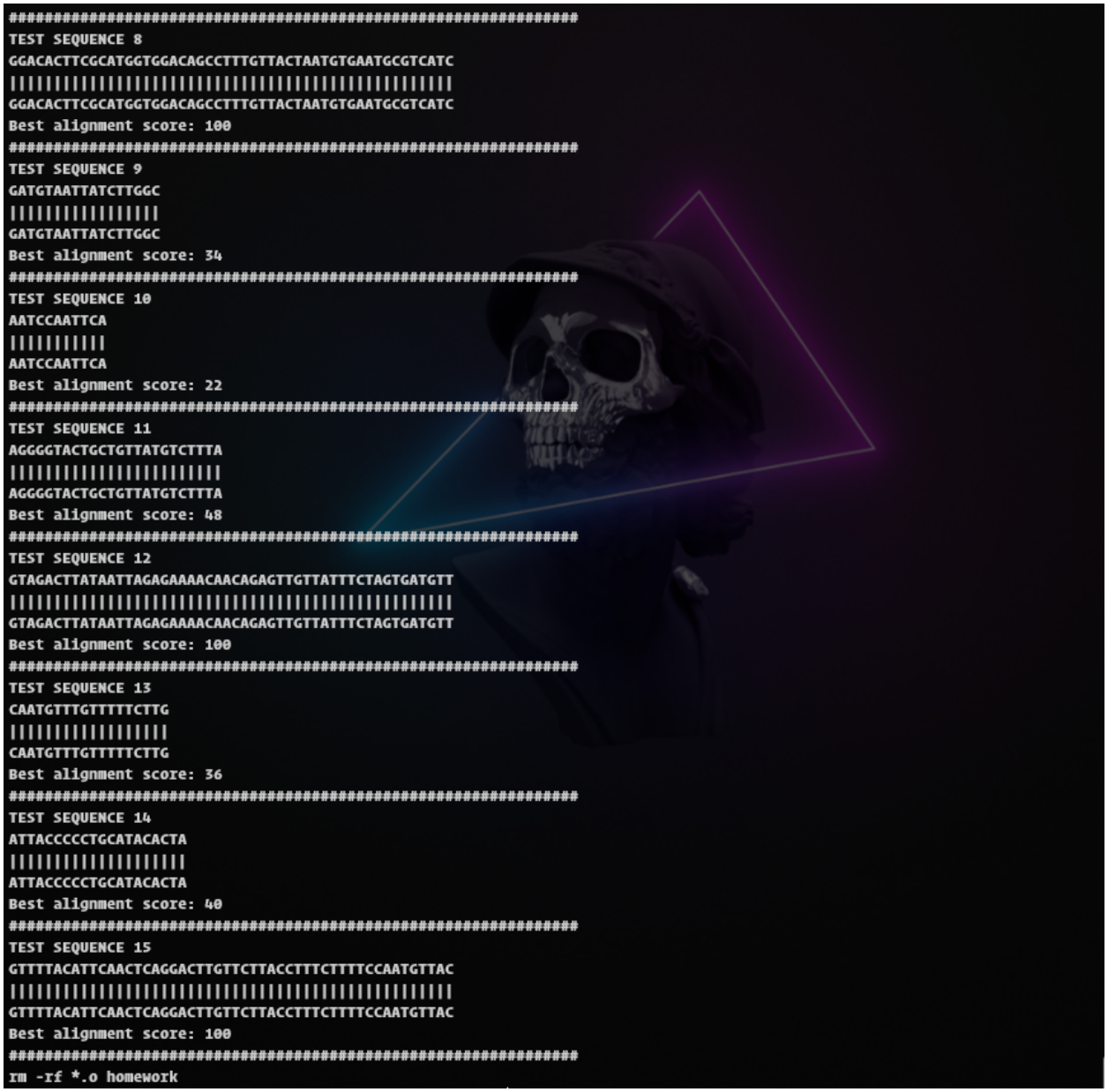
I have created a blast class that initializes the hash table of size 10000, reads the subject sequence, and stores the 11-mers sequences broken down from the subject sequence. Then, It takes the query sequence, breaks it down into 11-mers sequence, and performs the seed-based smith waterman algorithm.

I have expanded only one seed in the blast algorithm as suggested in the question. Also, the edge cases are handled.

After testing the code on the 50-mers provided below in Appendix B, we get the result as shown in the screenshot below.



```
[mp2525@wind ~/large-scale-data-structures/homework4 ]$ cat /scratch/mp2525/blast_q2a.txt
g++ -c homework.cpp -O3 -std=c++11
g++ -c smithWaterman.cpp -O3 -std=c++11
g++ -c blast.cpp -O3 -std=c++11
g++ -o homework homework.o smithWaterman.o blast.o -std=c++11
The number of arguments passed: 4
The first argument is: /home/mp2525/large-scale-data-structures/homework4/./homework
The second argument is: problem2A
The third argument is: ./sars_cov2.txt
The fourth argument is: ./test.txt
=====================================================
Initialized 29903 character sequences.
Broken down into 29893 11-mers sequences.
=====================================================
############################################################
TEST SEQUENCE 1
No Match found
############################################################
TEST SEQUENCE 2
ACAAGTGTGCC
||||||||||
ACAAGTGTGCC
Best alignment score: 22
############################################################
TEST SEQUENCE 3
No Match found
############################################################
TEST SEQUENCE 4
CAACACAACAA
||||||||||
CAACACAACAA
Best alignment score: 22
############################################################
TEST SEQUENCE 5
AACATGGCAAGGAA
||||||||||||||
AACATGGCAAGGAA
Best alignment score: 28
############################################################
TEST SEQUENCE 6
AAAGAAAATGACTCTAAAGAGGGTTTTTTCACTTACATTTGT
|||||||||||||||||||||||||||||||||||||||||
AAAGAAAATGACTCTAAAGAGGGTTTTTTCACTTACATTTGT
Best alignment score: 84
############################################################
TEST SEQUENCE 7
AGCTCTTGGAGGTTCCGTGGCTATAAAGATAAC
||||||||||||||||||||||||||||||||
AGCTCTTGGAGGTTCCGTGGCTATAAAGATAAC
Best alignment score: 66
```


##
TEST SEQUENCE 8
GGACACTTCGCATGGTGGACAGCCTTTGTTACTAATGTGAATGCGTCATC
||||||||||||||||||||||||||||||||||||||||||||||||||
GGACACTTCGCATGGTGGACAGCCTTTGTTACTAATGTGAATGCGTCATC
Best alignment score: 100
##
TEST SEQUENCE 9
GATGTAATTATCTTGGC
|||||||||||||||||
GATGTAATTATCTTGGC
Best alignment score: 34
##
TEST SEQUENCE 10
AATCCAATTCA
|||||||||||
AATCCAATTCA
Best alignment score: 22
##
TEST SEQUENCE 11
AGGGGTACTGCTGTTATGTCTTTA
|||||||||||||||||||||||||
AGGGGTACTGCTGTTATGTCTTTA
Best alignment score: 48
##
TEST SEQUENCE 12
GTAGACTTATAATTAGAGAAAACAACAGAGTTGTTATTTCTAGTGATGTT
|||||||||||||||||||||||||||||||||||||||||||||||||||
GTAGACTTATAATTAGAGAAAACAACAGAGTTGTTATTTCTAGTGATGTT
Best alignment score: 100
##
TEST SEQUENCE 13
CAATGTTTGTTTTTCTTG
||||||||||||||||||
CAATGTTTGTTTTTCTTG
Best alignment score: 36
##
TEST SEQUENCE 14
ATTACCCCCTGCATACACTA
||||||||||||||||||||
ATTACCCCCTGCATACACTA
Best alignment score: 40
##
TEST SEQUENCE 15
GTTTTACATTCAACTCAGGACTTGTTCTTACCTTTCTTTTCCAATGTTAC
||||||||||||||||||||||||||||||||||||||||||||||||||||
GTTTTACATTCAACTCAGGACTTGTTCTTACCTTTCTTTTCCAATGTTAC
Best alignment score: 100
##
rm -rf *.o homework

B. Test your code on a set of 1K, 10K, 100K, and 1M (million) completely random 50-mers, aligning them to the SARS-COV2 genome. How long did it take? Compare it to the results in problem 1B.

Ans:

```
[mp2525@wind ~/large-scale-data-structures/homework4 ]$ cat /scratch/mp2525/blast_q2b.txt
g++ -c homework.cpp -O3 -std=c++11
g++ -c smithWaterman.cpp -O3 -std=c++11
g++ -c blast.cpp -O3 -std=c++11
g++ -o homework homework.o smithWaterman.o blast.o -std=c++11
The number of arguments passed: 3
The first argument is: /home/mp2525/large-scale-data-structures/homework4/./homework
The second argument is: problem2B
The third argument is: ./sars_cov2.txt
=====================================================
Initialized 29903 character sequences.
Broken down into 29893 11-mers sequences.
=====================================================
Time to test BLAST with 1000 sequence: 0.025 seconds.
Time to test BLAST with 10000 sequence: 0.260 seconds.
Time to test BLAST with 100000 sequence: 2.658 seconds.
Time to test BLAST with 1000000 sequence: 26.691 seconds.
rm -rf *.o homework
```

The above screenshot shows the time taken to test the set of 1K, 10K, 100K, and 1M (million) completely random 50-mers, aligning them to the SARS-COV2 genome.

When comparing the results with the results in problem 1B, It is incredibly faster. It is because we do not have to do the computation of smith waterman on every broken query and subject sequence.

Task	Random Sequences	Execution time
1	1,000	0.025 seconds
2	10,000	0.260 seconds
3	100,000	2.658 seconds
4	1,000,000	26.691 seconds

C. Test algorithm's exhaustiveness. Randomly select 100,000 fragments from the SARS-COV2 genome and use these fragments to query the SARS-COV2 genome using the seed-based SW you implemented in part A. How many fragments were you able to find? Now introduce random errors into your 100,000 fragments at a 5% per-base error rate (every character has a 5% change of being changed to some other random character). Use these error-filled 100,000 fragments to query the SARS-COV2 genome again. How many fragments were you able to find?

Ans:

```
[mp2525@wind ~/large-scale-data-structures/homework4 ]$ cat /scratch/mp2525/blast_q2c.txt
g++ -c homework.cpp -O3 -std=c++11
g++ -c smithWaterman.cpp -O3 -std=c++11
g++ -c blast.cpp -O3 -std=c++11
g++ -o homework homework.o smithWaterman.o blast.o -std=c++11
The number of arguments passed: 3
The first argument is: /home/mp2525/large-scale-data-structures/homework4/./homework
The second argument is: problem2C
The third argument is: ./sars_cov2.txt
============================================================
Initialized 29903 character sequences.
Broken down into 29893 11-mers sequences.
============================================================
Total Random Sequences Found: 30155
Total Random Sequences Found (5% Error): 16255
rm -rf *.o homework
```

Randomly selecting 100,000 fragments from the SARS-COV2 genome with and without error-filled fragments and applying SW on the SARS-COV2 query, We get the following results.

Without error: 30155 fragments found
With a 5% error rate: 16255 fragments found

It can be inferred that adding error to the random generation algorithm increases the randomness of the sequence and tends to decrease match with the subject sequence.