ECE 551 Homework 1 (Spring 2016)
Due date is March 21th, 2016 beginning of class

1. Written (15 points) Consider the following 4 DNA sequences: GCAT, GCCA, GGCAT, GGCA. Find the optimal global alignment between all pairs with the following scoring function: match score is 3, mismatch score is -2, a linear gap model with a gap penalty of -2. Perform the multiple alignment of these four sequences using the Star Alignment technique. Choose the centre sequence that is most similar to the other sequences.

2. Written (20 points) Fill in the affine gap penalty matrix of AGC with ATGCC. Match score = 1, mismatch score = -1, gap opening penalty is -4 and gap extension penalty is -1. Give the score of the best alignment and list all the alignments with this score.

3. Written (25 points) Assume that we have the following reads:
ABCDEFGC
EFGCDHIJ
CDEFGCDH

a) Draw a de Bruijn graph for this data set with k-mer length 3. Edges should correspond to k-mers and nodes correspond to (k-1)-mers. Draw one weighted edge per distinct k-mer with weight equal to the number of times the k-mer.
b) Determine whether the graph is Eulerian or not.
c) Give an example of a walk through the graph that traverses three nodes and spells out a 4-mer that does not appear in any of the input reads. (This is an example where de Bruijn graphs may generate sequences that do not appear in reads.)

4. Programming (40 points)

a) Write a program that takes as input a set of reads and uses the greedy fragment assembly algorithm to output a single superstring that contains all reads as substrings. You must use the graph-based (Hamiltonian path) version of the algorithm. We will assume that 1) we are assembling a single-stranded sequence and 2) that no read is a substring any other read. The reads will be read from a file containing one read per line. To make this algorithm deterministic, you should have a specific rule for tie breaking. For two edges with the same weight choose the edge whose source node read is first in lexicographical order. If the source nodes are identical, then we choose the edge whose target node read is first in lexicographical order. You can try your algorithm with the reads in *test_reads.txt* to make sure it works correctly (It should output the superstring *the_quick_brown_fox_jumps_over_the_lazy_dog*).

b) Use your greedy assemble program to assemble a small subset of the reads (*ebola_reads.txt*) used to assemble the genome of an isolate of the Ebola virus, which caused a major epidemic in West Africa last year . Once correctly assembled, these reads form a short segment of the genome of this virus. To allow your assembler to succeed, the reads have been cleaned of errors and have have been oriented so that they all come from the same strand of the genome. Once you have assembled the genomic segment, use the BLASTX web service to search the NCBI database of proteins with your assembled sequence.