# EN 600.439/639: Computational Genomics

## Homework 4

Fall 2014, Prof. Ben Langmead

This homework is out of 40 points.

1. (5 pts) In lecture, we saw an example where Greedy-SCS collapsed a repeat. The input consisted of all length-6 substrings of `a_long_long_long_time`. The output was:

   ```
   a_long_long_time
   ```

   When we tried length-8 substrings instead, Greedy-SCS got the correct answer. Here's another input:

   ```
   to_every_thing_turn_turn_turn_there_is_a_season
   ```

   When I run Greedy-SCS taking length-6 substrings, the repeat is collapsed:

   ```
   to_every_thing_turn_turn_there_is_a_season
   ```

   When I try length-8 substrings, the repeat is *still* collapsed. Why doesn't using length-8 substrings fix the problem in this case? Would using length-9 substrings fix the problem? Why or why not?

2. We discussed De Bruijn graph assembly in class. Say we have the following input reads:

   ```
   ABCDEFGC
   EFGCDHIJ
   CDEFGCDH
   ```

   And the De Bruijn graph $k$-mer length = 3. Edges correspond to $k$-mers and nodes to $k$-1-mers.

   (a) (2 pts) Draw a De Bruijn graph for this data set. Draw one weighted edge per distinct $k$-mer, with weight equal to the number of times the $k$-mer occurs in the input.

   (b) (2 pts) Prove that the graph either is or is not Eulerian.

   (c) (2 pts) 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 of how De Bruijn graphs lack "read coherence".)

3. (5 pts) Give a linear (non-circular) string whose De Bruijn graph with $k$-mer length = 3 has exactly 4! = 24 distinct Eulerian walks. Two Eulerian walks are distinct if they visit edges in a different order. As your alphabet, use letters in the range $A$-$Z$ (but you don't need all of them).

   Hint 1: Solutions to this problem can have as few as 8 nodes, but it's fine if your solution has more.

   Hint 2: Your solution should have 2 semi-balanced nodes and the rest should be balanced.

4. (24 pts) Solve the problem here: `http://bit.ly/CG_UnpairedAsmChallenge`. Format your solutions as described on that page.

   (a) Submit your overlap graph as a file named `overlaps.txt`. Note that fields on a line should be separated by spaces.

   (b) Submit your unitigs as a file named `unitigs.txt`. Note that fields on a line should be separated by spaces.

   (c) Submit your assembled genome as a file named `solution.fa`. Output FASTA file should be formatted as described, with no more than 60 characters per line.

   Submit your code and include instructions on how to run your code to reproduce outputs (a) through (c) above.