Lab 1: Problem Set

This portion should be done with your lab partner. Your solution is due outside my mailbox by due date posted on the lab page. Part 2 of the lab is specified on the lab page.

Dynamic Programming Algorithms

The **minimum coin change** problem is as follows: given the denomination of coins for a currency and a particular amount of cents for change, produce the minimum number of coins necessary to make exact change. For example, if a customer pays a bill and expects 77 cents in change, the teller's task is to produce 77 cents using as few coins as possible.

For US currency (i.e., 1, 5, 10, 25), there is a known greedy solution that is correct. For example, to produce change for 77 cents, simply subtract the largest denomination (25 cents) three times, and then you are left with just two pennies for an answer of **5 coins**. For an arbitrary set of denominations, however, the greedy solution is not optimal so do not use it below.

For this problem, assume a simple scenario where there are three coins in the currency: **1, 3, and 5 cents**. Note that for this problem, you are trying to determine the number of coins in the optimal solution, not the actual coins needed.

1. Using a *recurrence relation* as we saw in class, define a simple recursive solution that is guaranteed to provide the optimal solution to the minimum coin change problem. Assume the change amount N and a function $minCoins(N)$ that returns the minimum number of coins needed to make change for N. HINT: if you start with 77 cents, there are three possible scenarios: use 1 cent (with 76 cents remaining), 3 cent (74 cents), or 5 cent (72 cents). You do not need to define the base case here.

2. Design an efficient dynamic programming solution for this problem using pseudocode. Your solution can be either top-down or bottom-up.

3. What is the run time of your solution in big-O notation?

4. Solve the min-coin change problem for 9 cents using your dynamic programming solution. Show the dynamic programming array for solving the problem with 9 cents and identify the number of coins needed.

Practice: Pairwise Sequence Alignment

5. Perform Needleman-Wunsch to solve the optimal global alignment with linear gap penalty on the sequences `AACGTTA` and `CGATAA`. Use a scoring function of gap penalty `-1`, and a substitution function of `+1` for matches and `-1` for mismatches. Your solution should clearly identify:

   - the final dynamic programming array including pointers
   - the optimal alignment score,
   - the traceback (i.e., alignment) for the optimal score using a *highroad* strategy to break ties.

6. For the above problem, perform Smith-Waterman for local alignment with linear gap penalty with the same sequences and scoring parameters. If there are multiple maximum alignments, return any of them.

Interpretation and Analysis

7. Amino acids D, E, and K are all charged molecules while V, I, and L are hydrophobic (i.e., afraid of water). Using the `BLOSUM50` scoring matrix (in your book or in lecture slides), determine:

   (a) the average substitution score within the group D, E, K (i.e., average of D to E, E to K, and D to K)

   (b) the average substitution score within the group V, I, L

   (c) the average substitution score *between* the two groups (i.e., the 9 pairs of D-V, D-I, D-L, E-V, E-I, and so on).

   (d) Interpret your results by explaining the biological significance of these three results. In particular, how the positive and negative values match our intuition about mutations. Keep your response to 2 sentences.

8. Complete this question once you have finished the programming assignment. Hemoglobin and myoglobin are evolutionary relatives. One major difference is that hemoglobin tends to form into tetramers (groups of 4) while myoglobin is a monomer.

   Despite this, they have similar functions - hemoglobin is used to transport oxygen while myoglobin stores oxygen. In humans, we can align the two proteins to see the significant amount of divergence using your implementation. Use your local alignment algorithm to align human hemoglobin beta `hemoglobin_ beta.txt` with human myoglobin, `myoglobin.txt`. How do your results differ as you increase the gap penalty? Specifically, measure the identity (exact matches) and gaps as you vary the gap penalty. Do the identity and number of gaps increase proportionally? Pick a gap penalty that provides good results and justify your choice in 1 or 2 sentences.

9. Analysis of research: You will received an email assigning you one of the research papers related to sequence alignment. Your job is to prepare for an in-lab discussion for lab on February 6 followed by a short write-up detailing your findings for your assigned paper:

   (a) Alignment of Whole Genomes by Delcher et al., *Nucleic Acids Research*, 1999.

   (b) Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs, *Nucleic Acids Research*, 1997.

   In lab discussion questions will be posted to Piazza under "Feb 6 Discussion Questions" Your paper should be 1-2 pages about any algorithmic component of your paper you chose. Specific ideas for what you can write about will be posted under "Lab 1: discussion paper questions". I am not looking for a summary of the entire paper - I am more interested in you being able to go in depth on a question.