Deterministic models and optimization

Programming projects

For the given project the tasks to complete are:

1. Write a text of **at most 6 pages** containing a discussion of the problem, the proposed algorithm, and a proof of correctness. The exposition should be clear, readable and well organized. Writing style and proper English usage will be taken into account for grading. The text should include all the references used in the project.
2. Write complete computer code of the algorithm, preferably in Python, R or C. **Each line of code must be commented** indicating its meaning in the flow of the algorithm. Check correctness of your algorithm on small examples.
3. Report the output of the algorithm on the data sets provided in the statement of the project.
4. The project must be submitted as a **single** PDF file containing:
   (a) Discussion of the problem and proposed algorithm. Must include: proof of correctness; analysis of complexity as a function of the input size; a brief discussion of the main algorithmic paradigm use in the solution: greedy, divide and conquer, dynamic programming. . .
   (b) References used in the project.
   (c) Complete computer code with comments.
   (d) Output of the algorithm on the data sets.

Submissions not following this format will not be considered.

Due: November 12 (strict deadline)

Edit distance

Given two strings of text X and Y, there we wish to measure by how much X and Y differ. Consider the following three operations on a string:

- D: Deletion of a character.
- I: Insertion of a character.
- S: Substitution of a character.

The edit distance $d(X, Y)$ is the minimum number of operations $\{D, I, S\}$ needed to perform on X to produce Y.

Tasks.

1. Design an algorithm that, given strings X and Y, computes the edit distance between X and Y. The algorithm should provide also the optimal sequence of operations transforming X into Y.
2. Modify the previous algorithm with a penalty cost function: operations D and I have unit cost 2, whereas operation S has unit cost 1.
3. Run both algorithms on the following pairs of input strings and report the edit distance and the minimal sequence of operations to transform one into another.
   (a) (DNA)
   X = ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA
   Y = TACTAGCTTACTTACCCATCAGGTTTTAGAGATGGCAACCA
   (b) (Proteins)
   X = AASRPRSGVPAQSDSDPCQNLAATPIPSRPPSSQSCQKCRADARQGRWGP
   Y = SGAPGQRGEPGPQGHAGAPGPPGPPGSDG
4. Estimate empirically the distribution of the edit distance (without penalty) between random words in the alphabet $\{A, B, C, D\}$ as follows. Generate N pairs of random words of length 50 in which each letter has the same probability 1/4 of being generated, and compute their edit distance. Plot the resulting histogram for $N = 100, 500$ and 1000, and estimate the expectation and variance from the randomly generated samples.

   Indicate how to modify the computation when the probabilities of the letters are not uniform but given by values (p_1, p_2, p_3, p_4) with $p_i \geq 0$ and $\sum_{i=1}^{4} p_i = 1$.