Scientific Computation
Spring 2022
Project 1

Due: Friday February 11th 4pm UK time

There are three main files for this assignment: 1) the one that you are reading which is the project description, 2) *project1.py*, a Python module which you will complete and submit on Blackboard (see below for details) and 3) *project1.tex*, a template file for your short report which will also be submitted on Blackboard. The discussion and figure(s) described below should be placed in this report. An example amino acid sequence for question 2 (*Sexample.txt*) has also been provided.

1. In this question, you will work with the function *code1* provided in *project1.py*. An N-element list of integers sorted in non-decreasing order should be provided as input; see the function header and documentation for further information.

   (a) (4 points) Provide a brief, clear description of the functionality of *code1*. Include a clear explanation of how the function's output is related to its input and the strategy the function uses to produce the output. A line-by-line description of the code is not needed.

   (b) (6 points) Analyze the "worst-case" cost of *code1*. Your analysis should include:

      (i) a clear and concise discussion of the theoretical running time and how it depends on N and N_0

      (ii) one or more well-designed figures illustrating key trends in the walltime required by the function

      (iii) a description of and explanation of the trends shown in the figure(s).

      The code that generates your figures should be placed in *test_code1*. Place your discussion and figure(s) in the appropriate section of your report. The __name__=='__main__' portion of the module should call *test_code1* and generate any figures included in your submission.

   Note: When answering (b) above, you may restrict N such that $N = 2^n$ where n is a non-negative integer. Similarly, you may constrain N_0 so its values are either 0 or of the form 2^m where m is a non-negative integer. When analyzing the walltime, you should not present results for an exhaustive range of values for N and N_0. Instead, you should use your theoretical analysis as guidance and carefully select values for N and N_0 so that key aspects of the efficiency of the function can be understood using a few figures. You do not need to consider values of N larger than ~ 1000 but are welcome to do so if it will help your analysis.

2. You will now develop code to analyze a length-n amino acid sequence S provided as input to the function, *findAA*. There are 20 naturally-occurring amino acids which will be labeled with the letters $a, b, c, ..., t$. One or more codons provide code for each of these amino acids. For example, GAC and GAT are code for d. The function *createTable* creates a dictionary which can be used to determine which amino acid any codon corresponds to. Note that there are three codons which do not correspond to amino acids. A list of p gene sequences, L_p, is also provided as input. Each gene sequence has length $3m$ where m is a non-negative integer. You are tasked with efficiently translating each gene sequence into a sequence of m amino acids and finding all locations in S of each of these length-m sequences.

   (a) (6 pts) Complete the function *findAA*, so that it efficiently finds all locations within S of the p amino acid sequences encoded in L_p. These locations should be stored in a length-P list, L_out, where the ith element of L_out is a list contains the locations of the ith input amino acid sequence. For example, if $S = aidddib$, and $L\_in = [\text{ATAGAC}, \text{GACGAC}]$, then L_out should be $L\_out = [[1][2, 3]]$. If any of the codons in the ith gene sequence do not correspond to an amino acid, then the code should set $L\_out[i] = [-1000]$. If the ith amino acid sequence is not found in S, set $L\_out[i] = [\ ]$. You should design your code for input with $n \gg m$, $m \gg 1$, and $p \gg 1$, though it may be helpful to consider smaller problem sizes when developing and testing your code. Your code should be efficient with regards to running time and memory usage, and you should assume that the cost of Python integer arithmetic is independent of the length of the integer. See the function documentation for further details on the function output.

   (b) (4 pts) Add a brief description of your code along with a clear and concise analysis of its running time to your report. Include an explanation of why your code should be considered to be 'efficient'. You do not need to run timing tests or discuss the wall time required by your code.

Further guidance

- You should submit both your completed python file and a pdf containing your discussion and figure(s). You are not required to use the provided latex template, any well-organized pdf is fine. To submit your assignment, go to the module Blackboard page and click on"Project 1". There will be an option to attach your files to your submission. (these should be named *project*1.*py* and *project*1.*pdf*). After attaching the notebook, submit your assignment, and include the message, "This is my own work unless indicated otherwise." to confirm the work as your own.

- Please do not modify the input/output of the provided functions without permission, and please do not import any modules without permission. You may create additional functions as needed, and you may use any code that I have provided during the term.

- Marking will be based on the correctness of your work and the degree to which your submission reflects a good understanding of the material covered up to the release of this assignment. Excluding figures, you should aim to keep the pdf version of your report to less than 2 pages.

- Open-ended questions require sensible time-management on your part. Do not spend so much time on this assignment that it interferes substantially with your other modules. If you are concerned that your approach to the assignment may require an excessive amount of time, please get in touch with the instructor.
- Questions on the assignment should be asked in private settings. This can be a "private" question on Ed (which is distinct from "anonymous") or by arrangement with the instructor.
- Please regularly backup your work. For example, you could keep an updated copy of your files on OneDrive.
- In order to assign partial credit, we need to understand what your code is doing, so please add comments to the code to help us.
- You have been asked to submit code in Python functions, but it may be helpful to initially develop code outside of functions so that you can easily check the values of variables in a Python terminal.