Assignment 0

CS329e - Elements of Software Design
DNA Sequences
(100 points)
No Grading for this assignment - Just for your practice

1 Description

DNA or deoxyribonucleic acid is a nucleic acid that contains genetic information. It is responsible for propagation of inherited traits. DNA is organized as two complementary strands that Watson and Crick called the Double Helix. Each strand is built out of nucleotides called bases of which there are four - adenine (A), thymine (T), cytosine (C), and guanine (G). The bases of the two complementary strands that make up the DNA pair up in this order: A+T, T+A, C+G, and G+C. Strands have directionality and the sequence order does matter. Genetic information is determined by the sequence of bases along the strand.

DNA has played an important role in research in computer science. For example research in string searching algorithms has been motivated by finding sequences in DNAs. For the present assignment, we are interested in finding the longest common base sequence in two DNA strands. Each strand is represented by the sequence of letters A, T, C, and G. For the two strands ACTG and TGCA the longest common sequence is TG. It is quite possible for two strands not to have any common sequence (a sequence of 1 base does not count). Also there could be two or more common sequences that have the same longest length.

In this assignment your task is to implement a python program with the name **dna.py**. You program should have the following input and output

Input:

You will read your data from a text file called dna.in from the command line.

```
python3 dna.py < dna.in

On Windows:

python dna.py < dna.in
```

The first line of data is an integer number n that gives the number of pairs of DNA to follow. You will read one pair of DNA strings at a time. The maximum length of each string is 80 characters. Assume that each string consists only of characters ’A’, ’T’, ’C’ and ’G’. It is acceptable if a string is in lower case or is in mixed upper and lower case. Convert both strings to upper case.

The file ”dna.in” contains the following plain text content:

```
3
GAAGGTCGAA
CCTCGGGA
ATGATGGAC
GTGATAAGGACCC
AAATTT
GGGCCC
```

Output:

Print out the longest common sequence(s) for the two strings. If there is more than one longest common sequence then print each of those sequences on separate lines in alphabetical order. Use the built-in sort function for lists. The sequences should be left aligned. Leave a blank line between the output of each input pair. There should be a blank line at the end. If there is no common sequence your program should output No Common Sequence Found.

Sample output session would look like:

```
TCG

GGAC
TGAT

No Common Sequence Found
```

Your program should have a good, clean logical structure. We will be looking at good documentation and descriptive variable names. You will adhere to the standard coding conventions in Python. Your file dna.py will have the following header:

```python
#  File: dna.py

#  Description:

#  Student Name:

#  Student UT EID:

#  Partner Name:

#  Partner UT EID:

#  Course Name: CS 313E

#  Unique Number:

#  Date Created:

#  Date Last Modified:

# Input: s1 and s2 are two strings that represent strands of DNA
# Output: returns a sorted list of substrings that are the longest
#         common subsequence. The list is empty if there are no
#         common subsequences.
def longest_subsequence (s1, s2):

def main():
  # read the data

  # for each pair
    # call longest_subsequence

    # write out result(s)

  # insert blank line

if __name__ == "__main__":
  main()
```

Extra Credit (5 pts)

If there are two or more largest DNA sub-strands that are identical, then output only one.

You may not change the names of the functions listed. They must have the functionality as given in the specifications. You can always add more functions than those listed.

Pair Programming

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming[1] and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will be submitting the code. But make sure that your partner's name and UT EID are in the header. If you are working alone then remove the partner's name and eid from the header.

1.1 Turnin

Turn in your assignment on time on Gradescope system on Canvas. For the due date of the assignments, please see the Gradescope and Canvas systems.

1.2 Python Code Style Format

Your code format style should be according to the PEP8 recommendation. Read more about PEP8 here

`https://peps.python.org/pep-0008/`

10% of your grade is for code style format and our grading system will reduce points for any violations.

1.3 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between ”cheating” and acceptable and beneficial interaction between students (In different assignment groups). Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way – visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **”two line rule”**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the ”two line rule” inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

[1] Read this paper about Pair Programming `https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF`

We will use the following Code plagiarism Detection Software to automatically detect plagiarism.

- Staford MOSS

  `https://theory.stanford.edu/~aiken/moss/`

- Jplag - Detecting Software Plagiarism

  `https://github.com/jplag/jplag` and `https://jplag.ipd.kit.edu/`