BDA350- Final Project

(This project is out of 20 marks and is worth 10% of your final grade)

Due date: Friday December 9th at Midnight

You may work with a partner to submit your final project. If you are planning to work with a partner, please let me know (via email or MS teams) who you are working with as soon as possible.

Copyright and Academic Integrity:

> Unauthorized use, copying or distribution of any course materials is strictly prohibited and students doing so may be in violation of Seneca's Intellectual Property Policy, privacy legislation (see Seneca's Freedom of Information and Protection of Privacy Policy), and the Canadian Copyright Act (see Seneca's Copyright Policy), will be reported to Student Conduct Office and may be subject to disciplinary actions as appropriate. Any unauthorized use, copying or distribution of any course materials which may aid yourself or assist others in cheating and/or plagiarism and is in violation of the Academic Integrity Policy will be reported to the Academic Integrity Committee.

Deliverables

Upload your python file as well as reflection.txt on blackboard.

Anagram analyzer:

An **anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once[1]. For example, the word *binary* can be rearranged into *brainy* and the word *adobe* into *abode*. In this project, we only focus on single word anagrams; we do not consider phrases or sentence anagrams.

Dataset

In this project, you are given a text file named **`adventures_of_huckleberry_finn.txt`**.

The text file is a novel by American author Mark Twain published in 1884[2].

[1] https://en.wikipedia.org/wiki/Anagram

[2] https://en.wikipedia.org/wiki/Adventures_of_Huckleberry_Finn

Instructions

Part 1: [70 marks]

Process the given dataset and find all the anagrams as well as their frequencies. Note that time complexity is a major factor in this program. Use the most suitable data structure that provides you with the lowest time complexity (linear or constant time). Your output should look like the following table:

Anagram Finder	
Anagrams	Frequency
Below, elbow	10
Night, thing	26

Table 1: Sample output

Write your code in a python file. In your code, include a docstring part. In this part, write your name, student number, and a brief description on how your code works.

Part 2: [15 marks]

Upgrade your code to include a search function. The search function takes a word as input and searches through your data structure to find the word or its anagram. If the word or its anagram is found, the function prints the word as well as its anagrams and its frequency. If the string is not found, the function prints an informative message.

Part 3: Reflection.txt [15 marks]

In a few sentences answer the following questions:

- What data structure did you use to write this project? Why?
- What is the time complexity of your code for part 1? How can you improve it?
- What is the time complexity of your code for part 2? How can you improve it?
- Write a paragraph about what you have learnt in this project, and what you found most challenging.