Task Specification

Task 1:

Task 1.1 Remove duplicated elements with a special rule

You are requested to develop an algorithm to read in a large amount of numbers, remove duplicated numbers with a bespoke rule, and write it back to a new file. However, your algorithm **must** follow the special rule to determine which unique element will be reserved.

Rule:
The middle element in the sequence of the duplicated elements can be retained only. For example, the list like [1, 2, **1**, 3, 1] is processed and its output should be [2, **1**, 3] because there are three '1' in the list, and the red one can be retained as it's the middle one, and the first and third ones are removed. If the number of the duplicated elements is even, the right one over the middle is retained. For example, the output of [1, 2, 1, 3, **1**, 1] is [2, 3, **1**] because there are four ones, and the third (or red) one is the one just over the middle.

There are more examples as follows to help you to understand the rule. The red elements in the lists are the elements which should be retained.

[2, 1, 2, 3, 1] -> [2, 1, **2**, **3**, **1**] -> [2, 3, 1]
[3, 2, 1] -> [**3**, **2**, **1**] -> [3, 2, 1]
[1, 2, 3, 3, 2, 1] -> [1, 2, 3, **3**, **2**, **1**] -> [3, 2, 1]
[3, 2, 1, 1, 2, 3, 2, 1, 3, 2] -> [3, 2, 1, **1**, 2, **3**, **2**, 1, 3, 2] -> [1, 3, 2]

You should design an efficient algorithm to implement this removing operation or pick-up operation with the consideration of fast execution and less memory usage. The higher mark is rewarded in terms of how efficient your program is and how much memory the program uses (less is better).

A testing file with massive numbers is provided for testing the speed of the execution.

Task 1.2: Data operations

We are implementing some basic data operations such as search, insertion and deletion. However, efficiency or execution speed is the only marking criterion.

A file with massive integers is provided. You are required to develop a program to read in all these numbers from the file, sort it and store it in a proper data structure.

Another operating file is provided, which consists of a sequence of data operations. Each operation includes a pair of numbers where the first number denotes which operation and the second one is the value to be processed. For operations, 1 denotes search, 2 is insertion and 3 is deletion. For example, (1, 10) is to search whether 10 is

included in the imported numbers, (2, 10) is to insert 10 as the ascending order, and (3, 10) means the deletion of 10 from the data structure. Note that for the deletion operation, if there are duplicated numbers, the deletion should remove all the duplicated numbers.

In short, this task includes the following requirements.

1. Import the integers from the provided file.
2. Sort these numbers in an ascending order.
3. Import the data operations from the provided file.
4. Implement these data operations in order.

The execution time should be calculated as the proof of the efficiency.

Task 1.3: Search with parallelism

You need to design a frequency-counting program with parallelism programming. Again, you are given two files: one is the file with a long text, and another is a file including a set of names. Your task is to count the frequency of each name in the text. For example, the result is like 'Harry Potter' 102, 'Ron Wesley' 54 or so on.

The result which contains all the names and their frequencies should be saved in a new file. The program must be a parallelized program.

Task 1.4: Cheapest train tickets

The national rail company requests you to design a train ticket search system. Simply speaking, users can input the names of the departure and destination, and the system returns the cheapest train ticket and all the station names of the route.

The company provides a csv file (railway_network.csv) which includes the direct connections of the adjacent stations and their weights (costs). For example, the tuple (Bristol Temple Meads, Bristol Parkway, 10) denotes that the cost between the two stations is 10 for the either way. This data is based on the existing national rail network that is provided as a PDF map. You can pick up any two station names from the map to test your program.

Since there are too many stations in London, we merge all these stations into **ONE** station called 'London'. That is, London is a hub to connect all lines from different directions.

The requirements of the task are as follows:

1. Import the csv file.
2. Input two station names for departure and destination.
3. Return the cheapest cost of the two stations and all the station names on the route.

Task 2

You should produce two mini reports and one short video to demonstrate your programs.

Task 2.1

Explain your design for Task 1 such as the choice of data structure and the logic of your algorithms. You can use pseudocode or workflow diagrams to illustrate your algorithms if necessary, and a pure and clear narrative is also accepted as long as you can make readers understand your design with ease.

You need to focus on the key algorithms only and don't need to explain any well-known algorithm or secondary procedures. For example, if you use merge sort or binary search, you just mention the algorithm names as we all know what these algorithms are. If you import data from a csv file, you don't need to explain how the program reads it line by line.

You also need to justify your design choices in terms of effectiveness, efficiency or memory spaces of the system that you have produced. Please check the marking criteria carefully for each task.

You should mention any unoriginal work. For example, if you use any Python library, you should address it in this task. If you use any significant code from open-source projects, you should remark it. Please feel free to use the code from the lab exercises.

The general discuss should be less than **500 words** except for the pseudocode and diagrams.

Task 2.2

Discuss any issue relating to computational sustainability that system developers should consider. Please use Task 1.4 to discuss the sustainability issue for the national railway services. Please give the reference if you use any formal definitions. This discuss should be less than **500 words.**