COMP2396 Object-oriented Programming and Java
Assignment 1 – Java Problem Solving

Due date: 25th September, 2025 23:59

Story

Handsome Jeff works as an intern at the IT department of Big Northwest Bus Company (BNB), the biggest public transport service provider in Heung Shing. His boss, Super Chim, asked him to write several Java programs to assist the daily operations of the company's bus fleet. However, he does not know Java and calls you for help. Help Jeff to solve the tasks below such that he won't be fired!

Question 1 – How Much Should I Pay? [50%]

Background:

The government has just constructed a bus-bus interchange facility at Tuen Mun Road. Passengers can take a bus route from anywhere to the facility (i.e., 1st trip), then interchange another bus route to reach their destination (i.e., 2nd trip).

To encourage passengers interchanging routes that BNB operates in this facility, the

company has set up the following discount scheme for passengers who pay both trips by Octopus card (an e-payment method):

1. For two BNB trips, if the 2nd trip's full fare is higher, the passenger pays only the difference otherwise, the 2nd trip is free.

2. If the route name taken in the 1st trip starts with "P", e.g., P960, the 2nd trip will be free regardless of the route the passenger takes.

3. The above discount scheme does not apply in the following situations:

   (i) The bus route taken in the 1st trip is not operated by the Big Northwest Bus Company.
   (ii) The route name taken in the 2nd trip starts with "A", e.g., A33X.
   (iii) The remaining balance in the passenger's Octopus card is less than HK$1 when paying for the 2nd trip.

   For situation (i) and (ii), full fare must be paid for the 2nd trip.
   For situation (iii), the payment fails and passenger has to pay the 2nd trip by cash.

Task:

Write a program Question1.java that calculates the remaining balance of the passenger's Octopus card, after he/she uses it to conduct payment for the 2nd trip on any BNB's bus routes.

The program should take in three lines of input with the following format:

<Remaining balance after 1st trip>
<Route name in 1st trip> <Company Code> <Full fare for 1st trip>
<Route name in 2nd trip> <Company Code> <Full fare for 2nd trip>

The program should output the remaining balance of the passenger's Octopus card after paying for the 2nd trip.

Assumptions:

1. The company code of the Big Northwest Bus Company is "BNB".

2. Length of the route name and the operating company code will not be longer than 4 characters. All English characters involved are capitalized.
3. The remaining balance and full fares of the bus trips must be an integer.
4. All passengers must take the 1st bus trip to enter the interchange facility, and all of them pay the 1st trip successfully by using their Octopus card.
5. There is no other purchase paid by the Octopus card in-between the two trips.
6. The three lines of input are always valid.

Sample Test Cases:

Test Case 1
Input:
8
58M BNB 9
960 BNB 14

Output:
The remaining balance is 3.

Test Case 2
Input:
2
962X CTB 23
961 BNB 14

Output:
The remaining balance is -12.

Test Case 3
Input:
-3
59M BNB 19
961 BNB 24

Output:
The remaining balance is -3.

Question 2 – Intelligent Traffic Avoidance [50%]

Background:

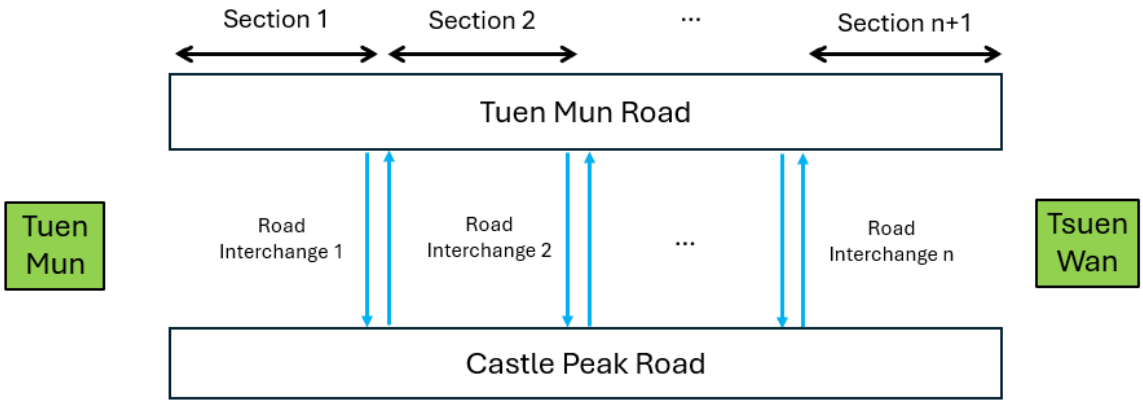
Tuen Mun Road and Castle Peak Road are two highways connecting Tuen Mun and

Tsuen Wan. Tuen Mun Road is infamous for 24-7 traffic jams and traffic accidents.

Suppose there are *n* road interchanges that connect the two highways (It makes sense that the number is a variable as some interchanges may be closed, say, for maintenance). The interchanges must be located in between the mid-sections of both highways. Drivers can swap to another highway with a time cost when they encounter any road interchanges. The time cost for road swapping is fixed and is the same for all the interchanges.

With a given set of time costs to travel between the n + 1 sections of both highways, a bus driver may compare the time cost for passing different sections of both highways, then figure out a path which results in the minimum time to travel from Tuen Mun to Tsuen Wan.

See the diagram below for illustration:



Task:

Write a program Question2.java that calculates the minimum time needed for the bus to travel from Tuen Mun to Tsuen Wan.

The bus may start the journey at the beginning of either highway. The bus arrives Tsuen Wan when it reaches the end of either highway.

The program should take in 4 lines of input with the following format:

<n>
<Time cost to switch to another road>
<Time to pass the 1st section of Tuen Mun Road> … <Time to pass the (n+1)th section of Tuen Mun Road>
<Time to pass the 1st section of Castle Peak Road> … <Time to pass the (n+1)th section of Castle Peak Road>

The program should output the minimum time needed for the bus to travel from Tuen Mun to Tsuen Wan.

Assumptions:

1. The bus can only go forward, but not backward.
2. The range of n will be at least 3 but not bigger than 20.
3. All inputs are positive integers and they must be valid.

The following is a pseudo code you can follow to aid you in programming the algorithm

```
ALGORITHM CalculateMinimumTime
    INPUT: n, switchCost, tunMunRoad[0..n], castlePeakRoad[0..n]
    OUTPUT: Minimum time needed

    // Initialize DP arrays
    DECLARE dp1[0..n], dp2[0..n]

    // Base cases
    dp1[0] ← tunMunRoad[0]
    dp2[0] ← castlePeakRoad[0]

    // Fill DP tables
    FOR i ← 1 TO n DO
        // For Tuen Mun Road
        stayOnTuenMun ← dp1[i-1] + tunMunRoad[i]
        switchToTuenMun ← dp2[i-1] + switchCost + tunMunRoad[i]
        dp1[i] ← MIN(stayOnTuenMun, switchToTuenMun)

        // For Castle Peak Road
        stayOnCastlePeak ← dp2[i-1] + castlePeakRoad[i]
        switchToCastlePeak ← dp1[i-1] + switchCost + castlePeakRoad[i]
        dp2[i] ← MIN(stayOnCastlePeak, switchToCastlePeak)
    END FOR

    // Return minimum of both final positions
    RETURN MIN(dp1[n], dp2[n])
END ALGORITHM
```

Sample Test Cases:

Test Case 1

Input:

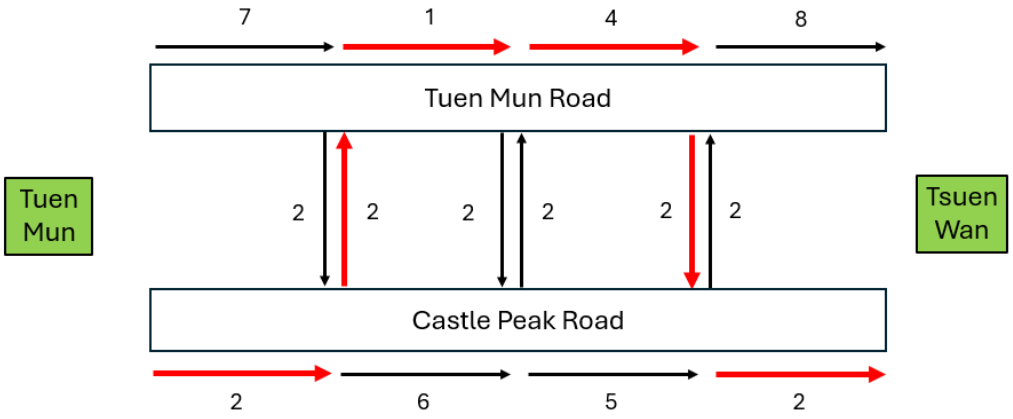
3

2

7 1 4 8
2 6 5 2

Output:
The minimum time needed is 13.

Explanation:
Refer to the diagram below. The red arrows indicate the path that the bus driver should take. The total journey time is 2 + 2 + 1 + 4 + 2 + 2 = 13.

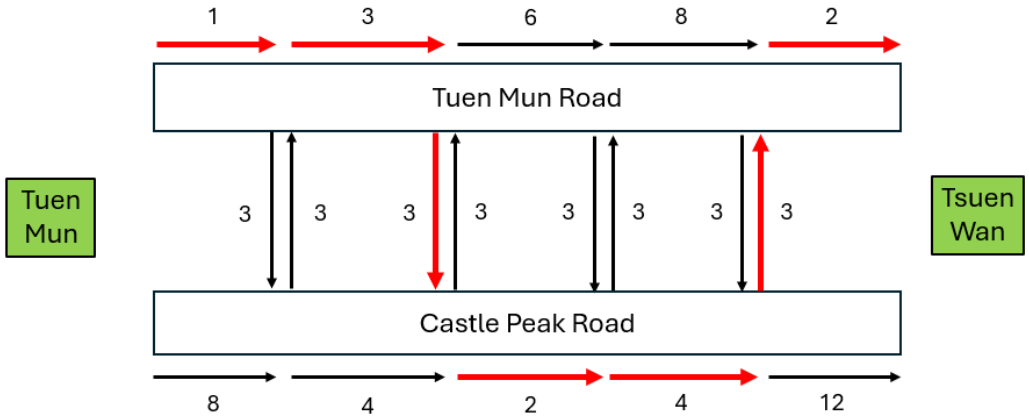


Test Case 2

Input:
4
3
1 3 6 8 2
8 4 2 4 12

Output:
The minimum time needed is 18.

Explanation:
Refer to the diagram below. The red arrows indicate the path the bus driver should take.
The total journey time is 1 + 3 + 3 + 2 + 4 + 3 + 2 = 18.



Grading & Submission

Marking
For each question,

100% marks are given to the **functionality** of your program.

- You may assume that the program input is always valid.

- You program output must exactly match to what is described in this document. Any mismatch, such as additional or missing space character, will lead to 0 mark.

You do not need to write Javadoc for this assignment.

You will get 0 mark if:

- You submit .class files instead of .java source files, or
- You submit java source files that cannot be compiled, or
- You submit java source files that are downloaded from the Internet, or
- You submit java source files from your classmates, or
- You submit java source files from friends taken this course last year.

Submission & Evaluation

Please submit Question1.java and Question2.java to Moodle VPL and evaluate.

- Double check your submission. Please check the assignment page again after submission to see all files you have submitted.
- You must evaluate your program before the assignment due date to get marks. **Late submission / evaluation is not allowed!**
- You are encouraged to keep an optional GitHub private repo for the assignment as a record. Please refer to the guidelines on Moodle for more details.
- If you encounter any technical problems, please contact us as soon as possible.
- We will use more test cases to test your program outputs against standard outputs after the submission deadline.

Usage of LLM

- Usage of LLMs is **strictly prohibited** for ALL questions in this assignment.
- All answers must be your own work. **When LLM is forbidden for a question, any form of consultation with LLM with regards to that question is treated as plagiarism.**

Wish you enjoy the assignment! Add oil!