**StructureFlow Ltd**

10 Queen Street Place, London

EC4R 1AG, United Kingdom

Our ref: Technical test

Date: 11 March 2022

Dear,

LIFT TEST

This programming challenge is to write a Test Driven Code, using JavaScript/Typescript (following MVC/OOP or built with nodejs/react) that manages 5 elevators in a building.

A 50 floors building (ground floor plus floors 1-50) has five elevators. To keep this simple, each elevator can move at 5M/second and each floor in the building is 3M high. When an elevator stops it takes 10 seconds from doors opening until they close. Elevators remain with their doors open at the floor until moved away. So if I got into an elevator on the ground (call it floor 0) and went to the top (floor 50) it would take 30 seconds to get to the top (=50 x 3/5) and 10 seconds. So after 40 seconds that elevator is available for use again. If I was on the 5th floor and the nearest elevator was on the 7th then it would take it 2 x 3/5 = 1.2 seconds to reach me and then 10 seconds to open, close and be ready to move off.

Instructions

You have to write a program which:

1. simulates the movement of the elevators.
2. reads in a JSON Response of requests and handles them.
3. Outputs a list of elevator movements and how long it took to move each person to their destination and keep a total of the total passenger time. The shortest overall total passenger time is the goal.
4. Code must contains tests

Time,from floor,to floor

Where time is in seconds from the start. Eg 567, and the two floor values are ints in the range 0 to 50. Examples 120,0,15 At time 120 a call to the ground floor to go to the 15th. 121,2,0 At time 121 a call to floor 2 to go to the ground.

GUIDELINES

Each request is a record in the JSON Response. (format: CSV, Each line has three comma separated numbers.)

1. Elevators start at random floors.
2. If an elevator is in motion it can answer calls in the direction it is moving in.
3. You can have uncalled lifts move to a default floor.
4. Each elevator is big so no problems about being full.
5. Don't worry about doors opening and closing, just treat it as a ten second delay.
6. Acceleration and deceleration are considered instantaneous. Ignore speeding up and slowing down.
7. The optimized code has the least total time to satisfy requests.
8. If a lift is descending it should not stop for a passenger who wants to go up or vice versa!
9. The journey time for the passenger stops when the elevator arrives at the desired floor.

The output should be a list of arrivals, something like this in seconds, to tenths of seconds accuracy.

* Time 4248.0 Floor 7 Passenger Pressed up (Going to Floor 10).
* Time 4248.0 Call Response to Floor 7. Elevator 1 on Floor 1. Ascending to Floor 7.
* Time 4251.6 Elevator 1 arrived Floor 7. Waited 10 seconds. Moving to Floor 10.
* Time 4263.4 Elevator 1 arrived Floor 10. Waited ten seconds.
* Passenger on Floor 7 - Total time 15.4, total all passengers is 346.8.

In this example an elevator on Floor 1 was sent straight away from the ground floor. It ascended 6 floors taking 6x3/5=3.6 seconds. It then waited ten seconds, then went to floor 10 taking 3 x 3/5 = 1.8 seconds. From pressing the button and getting there took the passenger 15.4 seconds.

Sincerely,

**Sebastien BRAMILLE**

Head of Engineering