Data Structures (Fall-2023)

Deadline: 8th Oct 2023, 11:59pm

Assignment# 02

Instructions:

- All the submissions will be done on Google classroom.
- You have to submit a cpp file named (21I-XXXX.zip). Naming convention has to be followed strictly. 30% marks will be deducted for not following submission guidelines.
- The student is solely responsible to check the final zip files for issues like corrupt files, viruses in the file, mistakenly exe sent.
- Be prepared for viva or anything else after the submission of the assignment.
- Zero marks will be awarded to the students involved in plagiarism.
- Late submission of your solution is not allowed. Submission within 30 minutes after the deadline will be accepted with 50% deduction. After that no submission will be accepted.
- Do not use Vectors in this Assignment. Usage of vectors will result in straight zero.
- Make sure you submit the correct code because test cases will be run on them.
- No part of the code should be commented.
- Understanding the assignment is also part of the assignment.
- All classes made must be templatized.

- **You need to combine all work in a single cpp file and submit.**

Multi-Modal Transport System for Fast University Students

Problem Statement:

Fast University is situated in the heart of Islamabad, with students hailing from different sectors of the city. They employ various modes of transportation for their daily commute, such as the metro, bus, or simply walking. However, due to the nature of the transportation systems, students often need to combine these modes. You can envision the city's locations as a numbered line starting from 1(one). Movement is restricted to forward and backward along this line. The metro is considerably faster than the bus, and the bus is faster than walking. Unfortunately, the metro can only stop at designated metro stations, and the bus can only stop at designated bus stops. Walking can be initiated from any bus stop and can end at any point.

To visualize these journeys, a special 3-layered linked list is employed, illustrated as follows:

Metro: 1----------3----------5-------6----------8----------
Bus: 1-------2----------4-------5----------7----------
Walking: 1-------2-------3-------4-------5-------6---------7-------8-------9-------10

The bottom layer represents walking, the middle layer represents the bus, and the top layer represents the metro. Each node in the layered linked list contains:

- An integer representing the location
- A reference `next` pointing to the next location within the current mode of transport
- A reference 'prev' pointing to the previous location within the current mode of transport.
- A reference `down` taking you down to the SAME location in a slower mode of transport.

For instance, in the diagram above, node 5 in the metro layer points to node 5 in the bus layer, which in turn points to node 5 in the walking layer. `next` will be set to null if there are no more locations to visit on the current mode of transport, and `down` will be set to null if you are in the walking layer or if the lower layer has no location with the same node id. prev will be set to null if there is no location before the current node in the same layer. You can directly go from the metro layer to the walking layer node if there is no bus node.

[Task 1 - Transport Layer Optimization: 70 marks]

You are tasked with designing a transport system that provides the following functionality.

Requirements:

- **addNode(int layer, int Location_id):** that adds a location in the respected layer, layers are numbered from 1-3.

- **DeleteNode(int layer, int Location_id):** that deleted a location in the respected layer.
- **GetPath(int start, int end):** returns a string with the optimal path. A path containing nodes of the first layer will always be better than the second layer, and a path containing nodes of the second layer will always be better than the third layer, i.e. the path in the highest layer if it exists will always be preferred.the path should contain all the nodes between start and end point. if start is **3** and end is **5**. then it must pass through node **4.**
- The person will always start their journey from the fastest mode available for starting node.

Input format:

The first line must contain three integers: **n**, **m**, and **q**, representing the number of stations in the first, second, and third layers, respectively.

- The second line should consist of **n** integers denoting the node IDs corresponding to the locations in the metro layer.
- The third line should consist of **m** integers denoting the node IDs corresponding to the locations in the bus layer.
- The fourth line should consist of **q** integers denoting the node IDs corresponding to the locations in the walking layer.

[Task 2 - Transport System with Cost Calculation : 30 marks]

In this transport system, we consider the cost of traveling between different locations . The primary goal is to compute and manage the cost of traveling between any two locations within the network. assume that all the nodes exist in all the transport layers.

Input format:

- The first line contains an integer **n**, representing the total number of locations in the transport network.
- The next three lines contain a series of **n-1** integers each, where each line corresponds to a different mode of transportation: metro, bus, and walking respectively. These lines represent the costs associated with traveling from one location to another. The cost of going from one node to the same node on another layer above or below it is 5. going from layer **1-3** or **3-1** will cost 10.
- The **i-th** integer in each line represents the cost of traveling from the **i-th** location to the **(i+1)-th** location using the respective mode of transportation.
- the person can start their journey from any layer on the starting node and end at any layer on the ending node.

Create a function **GetMinCost(int start , int end)** that returns the minimum cost for going from starting point to the end point.