

(The Underground) Assignment 1: Subway Maps

Total grade: 100

Due date: Monday, February 8, 2021 at 23:59 EST

Late assignments will be deducted 10% per day up to five days (including weekends)

Below is the iconic subway map of London (England). Although the map has evolved over the years, its fundamental and revolutionary design by Harry Beck has remained largely unchanged since it was first introduced to the British public in 1933. In graph theoretic terms, each subway station is a vertex, and each edge between a pair of stations is a subway connection. Each connection is coloured to represent its part of a subway line, and more than one connection may run between adjacent stations. For example, yellow and green connections run between Westminster and St. James's Park.



The subway map of London or any other subway system may be represented as a variation of the adjacency list implementation as shown below.

```
public enum Colour {RED, YELLOW, GREEN, ... }           // For example
class Node {
    public Station Connection {get; set;}                // Adjacent station (connection)
    public Colour Line {get; set;}                       // Colour of its subway line
    public Node Next {get; set;}                        // Link to the next adjacent station (Node)
    public Node (Station connection, Colour c, Node next) { ... } (2%)
}
class Station {
    public string Name {get; set;}                      // Name of the subway station
    public bool Visited {get; set;}                     // Used for depth-first and breadth-first searches
    public Node E {get; set;}                           // Linked list of adjacent stations
    public Station (string name) { ... }                 // Station constructor (2%)
}
class SubwayMap {
    private List<Station> S;                             // List of stations
    public SubwayMap( ) { ... }                         // SubwayMap constructor (2%)
    public int FindStation (string name) { ... }
    public void InsertStation (string name) { ... }      // Duplicate station names are not allowed (4%)
    public void RemoveStation (string name) { ... }      (16%)
    public void InsertConnection (string name1, string name2, Colour c) { ... } (8%)
    public void RemoveConnection (string name1, string name2, Colour c) { ... } (8%)
    ...
}
```

Task 1: Implement each of the methods above, keeping in the mind the following requirements.

1. Each subway connection (edge) is undirected.
2. Each subway connection is coloured as part of a subway line.
3. An adjacent pair of subway stations may have multiple subway connections differentiated by colour.
4. Each subway station (vertex) stores its connections (edges) to adjacent stations in a linked list.

Task 2: Using a breadth-first search, implement a method called **FastestRoute (from, to)** which outputs the shortest sequence of subway stops (including transfers) between two given stations. (16%)

Task 3: Using a depth-first search, implement a method called **CriticalConnections ()** which outputs each connection (edge) which breaks the subway map into two parts if it is removed. (16%)

Task 4: Implement a main program to create your own subway map and to drive your test cases. (10%)

Task 5: Draw your own connected subway map and prepare your test cases. Include the map, test cases, and test results in a separate .pdf test document. (12%)

Inline (Source Code) Documentation (4%)

Submit your project (all files including an executable) as well as your test document at myLearningSystem.