

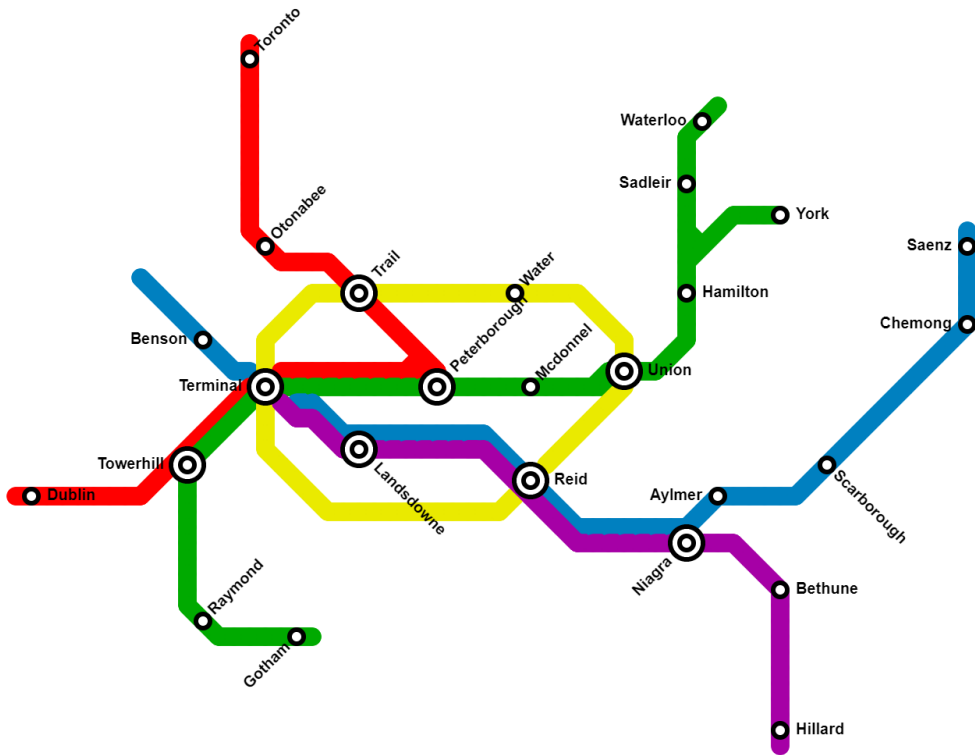
COIS 3020, Winter 2023

Subway Shortest Path

Contents

SubwayMap	2
Testing.....	3
Case 1: Inserting same station twice.	3
Case 2: Connection between non-existed stations.....	3
Case 3: Shortest route between two adjacent stations.....	4
Case 4: Shortest route to the same station	4
Case 5: Shortest route between two stations with multiple routes.....	4
Case 6: Shortest route between two stations with multiple routes of same distance.	5
Case 7: Shortest route between two far stations.	5
Case 8: Shortest route after removing a connection.....	5
Case 9: Shortest route after removing one connection from stations connected by multiple lines....	6
Case 10: Shortest route between two stations after removing a station in the route.	6
Case 11: Removing connection between two stations but with the wrong colour.....	7

SubwayMap



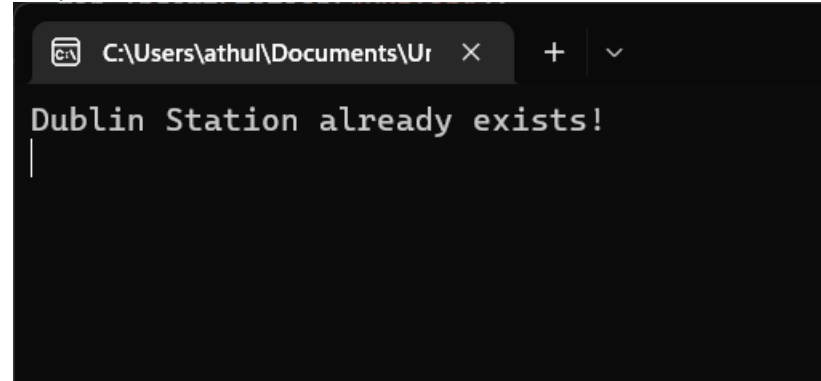
Testing

Case 1: Inserting same station twice.

Input:

```
map.InsertStation("Dublin");  
map.InsertStation("Dublin");
```

Output:

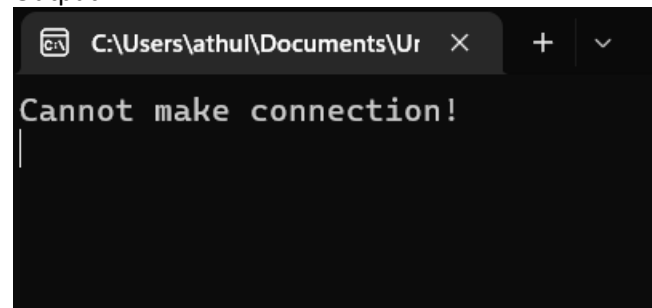
A screenshot of a Windows console window. The title bar shows the file path 'C:\Users\athul\Documents\Ur'. The console output displays the text 'Dublin Station already exists!' on the first line, with a cursor on the second line.

Case 2: Connection between non-existed stations

Input:

```
bool val;  
    val = map.InsertConnection("Oshawa", "Toronto", Colour.Red);  
if (!val)  
    Console.WriteLine("Cannot make connection!");
```

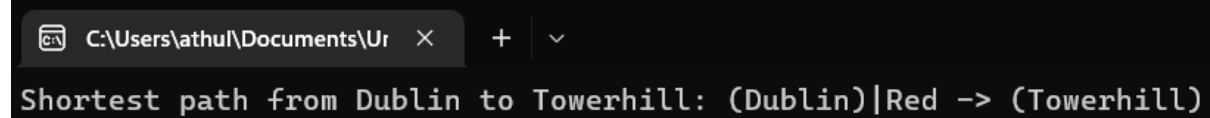
Output:

A screenshot of a Windows console window. The title bar shows the file path 'C:\Users\athul\Documents\Ur'. The console output displays the text 'Cannot make connection!' on the first line, with a cursor on the second line.

Case 3: Shortest route between two adjacent stations

Input:
`map.ShortestRoute("Dublin", "Towerhill");`

Output:



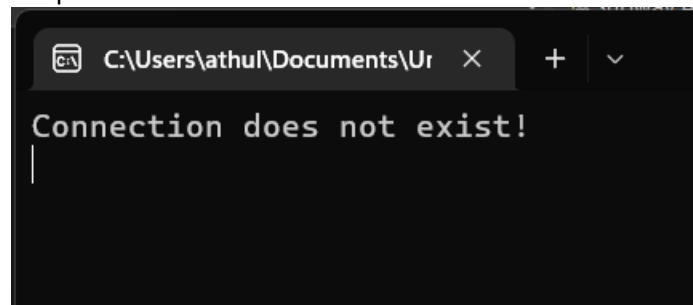
A screenshot of a terminal window with a dark background. The title bar shows the file path 'C:\Users\athul\Documents\Ur'. The terminal output displays the shortest path from Dublin to Towerhill as '(Dublin)|Red -> (Towerhill)'.

```
Shortest path from Dublin to Towerhill: (Dublin)|Red -> (Towerhill)
```

Case 4: Shortest route to the same station

Input:
`map.ShortestRoute("Dublin", "Dublin");`

Output:



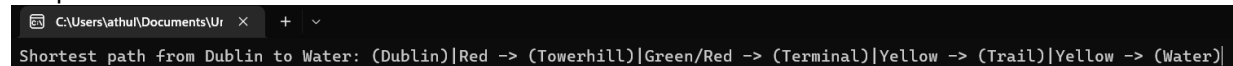
A screenshot of a terminal window with a dark background. The title bar shows the file path 'C:\Users\athul\Documents\Ur'. The terminal output displays the error message 'Connection does not exist!'.

```
Connection does not exist!
```

Case 5: Shortest route between two stations with multiple routes.

Input:
`map.ShortestRoute("Dublin", "Water");`

Output:




A screenshot of a terminal window with a dark background. The title bar shows the file path 'C:\Users\athul\Documents\Ur'. The terminal output displays a complex shortest path from Dublin to Water, involving multiple intermediate stations and line changes.

```
Shortest path from Dublin to Water: (Dublin)|Red -> (Towerhill)|Green/Red -> (Terminal)|Yellow -> (Trail)|Yellow -> (Water)|
```

Case 6: Shortest route between two stations with multiple routes of same distance.

Input:
`map.ShortestRoute("Terminal", "Union");`

Output:

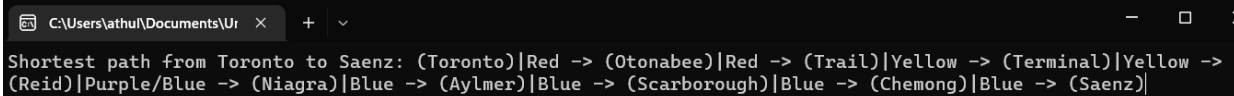


```
C:\Users\athul\Documents\Ur  X + v
Shortest path from Terminal to Union: (Terminal)|Yellow -> (Reid)|Yellow -> (Union)|
```

Case 7: Shortest route between two far stations.

Input:
`map.ShortestRoute("Toronto", "Saenz");`

Output:



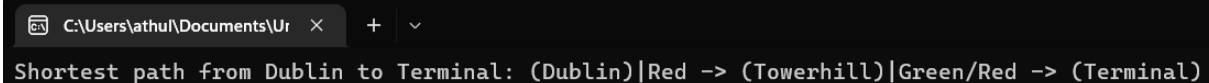
```
C:\Users\athul\Documents\Ur  X + v
Shortest path from Toronto to Saenz: (Toronto)|Red -> (Otonabee)|Red -> (Trail)|Yellow -> (Terminal)|Yellow -> (Reid)|Purple/Blue -> (Niagra)|Blue -> (Aylmer)|Blue -> (Scarborough)|Blue -> (Chemong)|Blue -> (Saenz)|
```

Case 8: Shortest route after removing a connection.

Input:
`map.RemoveConnection("Dublin", "Towerhill", Colour.Red);`
`map.ShortestRoute("Dublin", "Terminal");`

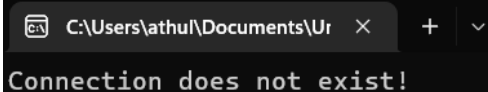
Output:

Before:



```
C:\Users\athul\Documents\Ur  X + v
Shortest path from Dublin to Terminal: (Dublin)|Red -> (Towerhill)|Green/Red -> (Terminal)
```

After:



```
C:\Users\athul\Documents\Ur  X + v
Connection does not exist!
```

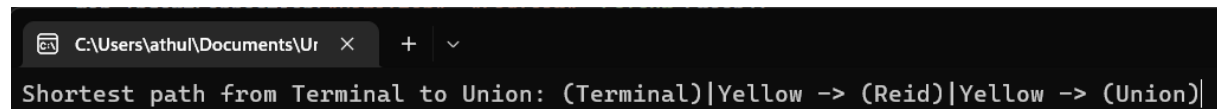
Case 9: Shortest route after removing one connection from stations connected by multiple lines.

Input:

```
map.RemoveConnection("Terminal", "Reid", Colour.Yellow);  
map.ShortestRoute("Terminal", "Union");
```

Output:

Before:



```
C:\Users\athul\Documents\Ur  X + v  
Shortest path from Terminal to Union: (Terminal)|Yellow -> (Reid)|Yellow -> (Union)|
```

After:



```
C:\Users\athul\Documents\Ur  X + v  
Shortest path from Terminal to Union: (Terminal)|Yellow -> (Trail)|Yellow -> (Water)|Yellow -> (Union)|
```

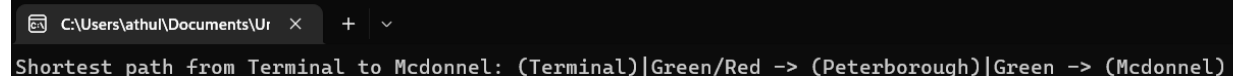
Case 10: Shortest route between two stations after removing a station in the route.

Input:

```
map.RemoveStation("Peterborough");  
map.ShortestRoute("Terminal", "Mcdonnel");
```

Output:

Before:



```
C:\Users\athul\Documents\Ur  X + v  
Shortest path from Terminal to Mcdonnel: (Terminal)|Green/Red -> (Peterborough)|Green -> (Mcdonnel)
```

After:



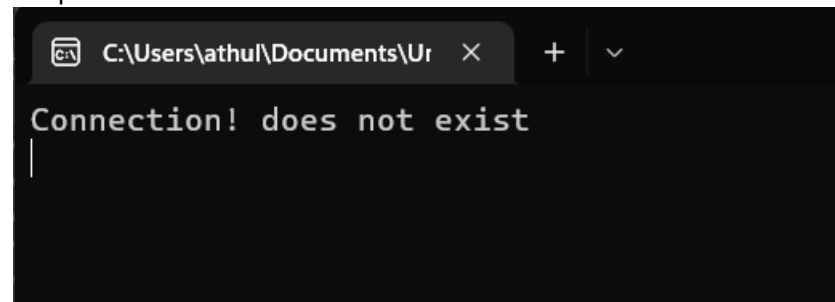
```
C:\Users\athul\Documents\Ur  X + v  
Shortest path from Terminal to Mcdonnel: (Terminal)|Yellow -> (Reid)|Yellow -> (Union)|Green -> (Mcdonnel)
```

Case 11: Removing connection between two stations but with the wrong colour.

Input:

```
bool val;  
    val = map.RemoveConnection("Trail", "Terminal", Colour.Blue);  
    if (!val)  
        Console.WriteLine("Connection! does not exist");
```

Output:

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Users\athul\Documents\Ur' with a close button. The command prompt displays the text 'Connection! does not exist' on a single line, with a cursor at the end of the line.

```
C:\Users\athul\Documents\Ur  
Connection! does not exist
```