

Lab 2: Shortest path under constraints

Start date: 07/06/2025. Due date: 15/06/2025.

Objective: In this lab, you will implement a program to find a shortest path under constraints.

Problem Statement: You are given a graph where each edge has a weight representing the time required to traverse that edge. Each vertex in the graph has a traffic light. Each traffic light operates on a fixed cycle time t , which is given in the dashed boxes.

When you arrive at a vertex with a traffic light, if your arrival time is **not divisible by t** , you must **wait** until the next time that **is divisible by t** before you are allowed to continue.

On the other hand, if you arrive at a time that is **divisible by t** , you are allowed to proceed immediately without waiting.

Using **Dijkstra's algorithm**, determine *the fastest route* from the source vertex A to the terminal vertex G, and report the **path taken** and the **total time**.

Requirements:

1. Input:

- The input file is “tests.txt”.
- An **undirected** graph is represented in a list of edges and weighted vertices.
- The first line contains the number of graphs.
- The second line contains the number of vertices in the first graph, says n .
- The next n lines contain the cycle times of the vertices.
- From the line $n+3$, each line contains an edge with its two endpoints and its time to travel along the edge.
- After reading the first graph, the second and remaining graphs start with a single line showing its number of vertices, followed by the cycle times of the vertices and the travel time along each edge.
- Example Input:
2
7
A, 2
B, 3
C, 5
D, 4
E, 6
F, 2
G, 1

A, B, 4
 A, C, 2
 B, D, 5
 C, D, 8
 C, E, 10
 D, F, 6
 E, F, 3
 F, G, 1
 7
 A, 2
 B, 3
 C, 4
 D, 2
 E, 5
 F, 3
 G, 1
 A, B, 2
 A, C, 4
 B, D, 3
 C, E, 5
 D, E, 1
 D, F, 2
 E, G, 3
 F, G, 4

2. Output:

- The output file must be name as “output_StudentID.txt”, where StudentID is your student ID.
- The outputs are written in a file with **2m lines**, where m is the number of graphs in the input file.
- Lines $2i - 1$ and $2i$ must output the shortest time and the path for the i th graph, respectively. Each vertex in the path must be separated by a space.
- What have been written in the output file must be also shown in the terminal.
- For example, for the input above, the following output must be written in file and shown in the running terminal:

19
 A B D F G
 13
 A B D E G

3. Tasks:

- Replace StudentID with your student ID.
- Your program must be a C++ program.
- The cpp file containing the main function must be named with your student ID, i.e., StudentID.cpp.
- By using the following command line in terminal or cmd (in Windows):
`g++ StudentID.cpp -o StudentID`
`StudentID tests.txt StudentID.txt (or ./StudentID tests.txt StudentID.txt)`

the output must be written in file “StudentID.txt” and shown in the terminal.

4. Evaluation Criteria:

- You are NOT ALLOWED to use any built-in data structure or library capable of handling shortest path.
- Correctness of results for various test cases.
- A fast and correct algorithm for large graphs.

Report guidelines:

1. Submit your source code and report.
The report must be in the pdf format and named as “StudentID.pdf”.
All source codes must be placed in a folder named “StudentID.zip”.
2. Write a report to describe what you have done to finish this assignment.
3. The first page of your report should contain your full name, student ID, and a table that describes which part of your source code is written by yourself and which part is referenced. For the reference part, you must explicitly write the reference.

No.	Percentage understood	Content understood	Percentage Referenced	Content Referenced	Reference Source
1	10%		90%		
2					

4. Any violation of this guideline will result in zero.