COMP2119 Introduction to Data Structures and Algorithms
Assignment 3 - Programming Challenge

Due Date: Nov 8, 2024 7:00pm

Background

You are Rain Carradine, one of the main characters in the movie *Alien: Romulus*. You and your friends are working as slaves on planet LV-410, one of the largest outer-space colonies built by the Weyland Company (aka "The Company").

One day, an abandoned secret spaceship appears near LV-410. To escape from the planet, you and your friends decide to break into the spaceship and steal the fuel essential for interstellar travel.

After scanning the structure of the spaceship, you find that it consists of n sectors. You also obtain information about the m accessible corridors between these n sectors, like which 2 sectors each corridor connects with, and the time t needed to travel between 2 sectors by using the corridor.

Complete the following tasks to help yourself surviving in the spacecraft and get the fuel you need!

Part A

Submit your answers for Part A only.

Task A [40%]
Suppose you are at spacecraft sector s now. Among the accessible corridors, you find that p out of n sectors of the spacecraft are out of reach from your sector. The spacecraft's onboard security system, *Mother*, blocked access from your sector to these p sectors with unknown reasons. To estimate how much hacking is needed for unlocking more corridors, you wish to find the value of p.

Complete the file **UniversityNumber-A.cpp** to calculate the answer.

Input Format:

- The first line has two integers n and m. n denote the number of spacecraft sectors. m denotes the number of accessible corridors connecting different sectors.
- Each of the next m lines contains three integers a, b, t. For a and b, they are the integer ids of the two sectors connected by the corridor. t denotes the time to travel between sector a and b.
- The last line contains an integer s, which denotes the sector that the characters are now at.

For all the input data, $1 \leq n, t \leq 100$ and $0 \leq m \leq 5000$.

You can also assume that the sector ids must start from 0 and ends at $n - 1$.

Output Format:
Output a single integer p, the number of sectors that is out of reach from s.

Sample Test Cases:

Sample Input	Sample Output	Explanation
7 4 0 1 2 0 2 3 3 4 2 5 6 1 0	4	There is no path from sector 0 to sectors 3, 4, 5, 6. Therefore, there are 4 sectors not accessible from sector 0.
5 6 0 1 2 0 2 3 1 2 1 2 3 4 2 4 6 3 4 8 2	0	One may go from sector 2 to all other sectors. Therefore, there are 0 sectors not accessible from sector 2.

Task B [30%]
After some time, you successfully hacked into *Mother* and gained access to more corridors. You now can access sector d, where the fuel is at. Due to the bizarre atmosphere present on the spacecraft, you wish to find the minimum time to travel from sector s (i.e. your current location) to sector d and get the fuel as soon as possible.

Complete the file **UniversityNumber-B.cpp** to calculate the answer.

Input Format:

- The first line has two integers n and m. n denote the number of spacecraft sectors. m denotes the number of accessible corridors connecting different sectors.
- Each of the next m lines contains three integers a, b, t. For a and b, they are the ids of the two sectors connected by the corridor. t denotes the time to travel between sector a and b.
- The last line contains two integers s and d. s denotes the sector that the characters are now at, and d denotes the destination sector.

For all the input data, $1 \leq n, t \leq 100$ and $0 \leq m \leq 5000$.

You can assume that the sector ids must start from 0 and ends at $n - 1$, and time starts at 0. You may also assume the characters spend negligible time within each sector.

Output Format:
Output a single integer *total*, the minimum time needed to travel from sector s to sector d. If there is no feasible path, output -1 as the answer.

Sample Test Cases:

Sample Input	Sample Output	Explanation
5 5 0 1 2 0 2 3 1 3 1 2 4 2 3 4 1 0 4	4	You are at sector 0 and wish to travel to sector 4. To minimize the total time of travel, the fastest way is to take the transitions [0 1 2], then [1 3 1], then [3 4 1]. Therefore, the total time cost is 2 + 1 + 1 = 4.
7 4 0 1 2 0 2 3 3 4 2 5 6 1 0 6	-1	You are at sector 0 and wish to travel to sector 6. However, there is no path from sector 0 to sector 6. Therefore, the program should output -1.

Task C [30%]
Suppose you have reached the fuel sector and obtained the fuel needed. During the fuel searching process, you guys realized that the spacecraft is an abandoned secret lab run by the Company for studying the behaviour of aliens. *Mother* previously locks the sectors mentioned in Task A to prevent the dangerous species from escaping into the outer space.

Suddenly, *Mother* gives a warning that some aliens are escaping. It initiates a security lockdown that each sector will be completely locked down after a specific time l (Each sector can have different values of l). One can neither enter or leave the sector equal to or after the above-mentioned time limit.

Suppose the spacecraft exit is located at sector d. To escape from the spacecraft in time, you wish to calculate the minimum time needed to travel from the current sector (i.e. sector s) to sector d under the above-mentioned constraints.

Complete the file **UniversityNumber-C.cpp** to calculate the answer.

Input Format:

- The first line has two integers n and m. n denote the number of spacecraft sectors. m denotes the number of accessible corridors connecting different sectors.
- Each of the next m lines contains three integers a, b, t. For a and b, they are the ids of the two sectors connected by the corridor. t denotes the time to travel between sector a and b.
- The next line contains n integers, each denotes the time l_i available before complete lockdown of the i^{th} sector.
- The last line contains two integers s and d. s denotes the sector that the characters are now at, and d denotes the destination sector.

For all the input data, $1 \leq n, t \leq 100$ and $0 \leq m, l_i \leq 5000$.

You can assume that the sector ids must start from 0 and ends at $n-1$, and time starts at 0. Also, you may assume the characters spend negligible time within each sector.

Output Format:
Output a single integer *total*, the minimum time needed to travel from sector s to sector d. If there is no feasible path, output -1 as the answer.

Sample Test Cases:

Sample Input	Sample Output	Explanation
5 5 0 1 2 0 2 3 1 3 1 2 4 2 3 4 1 5 1 6 10 12 0 4	5	You are at sector 0 and wish to travel to sector 4. The time limit for lockdown of sector 0-5 are 5, 1, 6, 10, 12. To minimize the total time of travel under the constraints, the fastest way is to take the transitions [0 2 3], then [2 4 2]. Therefore, the total time cost is 2 + 3 = 5.
5 5 0 1 2 0 2 3 1 3 1 2 4 2 3 4 1 5 1 6 10 3 0 4	-1	You are at sector 0 and wish to travel to sector 4. The time limit for lockdown of sector 0-5 are 5, 1, 6, 10, 3. However, there is no path such that you can travel from sector 0 to sector 4 within the time limit. Therefore, the program should output -1.

Submission & Grading

Assignment Completion

- You must use C++ to complete the assignment. The sample code is written in C++.
- We have provided a source file as a sketch for task A. You should write your code based on that for all the tasks.
- We are unable to assist with debugging their source code due to resource limitations. However, we can provide high-level hints and guidance.

Judging

- Your program will be judged by a computer automatically. A C++ compiler will be used.
- There will be 10 test cases for each task. Each test case worth 4 marks for Task A, while each test case worth 3 marks for Task B and C.
- For each test case, you get full mark if your program passes it. Otherwise, you will get 0 mark.
- The time limit for each test case is 1 second.

Important Grading Remarks

- Please follow the exact input and output format. Any mismatch, such as additional or missing space character, will lead to 0 mark.
- You should read from standard input and write to standard output. e.g, scanf/print, cin/cout.
- You may assume the grader's computer contains the following libraries:
  <iostream>, <string>, <vector>, <stack>,<queue>, <algorithm>, <math.h>, <limits>
- If your program cannot compile due to using a library that does not exist on the grader's computer, 0 mark will be given.

Self Testing

- You should test your program by yourself using the provided sample input/output file. The sample input/output is **different** from the ones used to judge your program, and it is designed for you to test your program by your own.
- Note that your program should always use standard input/output.
- To test your program in Linux or Mac OS X (this is the preferred option):
  1. Put your source code "main.cpp" and sample input file "input.txt" in the same directory.
  2. Open a terminal and go to that directory.

3. Compile your program by "g++ main.cpp -o main".
4. Run your program using the given input file, "./main < input.txt > myoutput.txt".
5. Compare myoutput.txt with sample output file.
6. Your output in myoutput.txt needs to be **exactly** the same as the sample output. Otherwise, it will be judged as wrong.
7. You may find the **diff** command useful to help you compare two files, like "diff -w myOutput.txt sampleOutput.txt". The -w option ignores all blanks (SPACE and TAB characters).

- To test your program in Windows (it's fine but maybe it's complicated to install the compiler):

1. Compile your program, and get the executable file, "main.exe".
2. Put sample input file, "input.txt", in the same directory as "main.exe".
3. Open command line and go to the directory that contains "main.exe" and "input.txt".
4. Run "main.exe < input.txt > myoutput.txt".
5. Compare myoutput.txt with sample output file. You may find "fc.exe" tool useful.
6. Your output in myoutput.txt needs to be **exactly** the same as the sample output. Otherwise, it will be judged as wrong.

- If these two options do not work, you can use an online compiler as a last resort: `https://www.onlinegdb.com/online_c++_compiler`

Manual Code Checking

We understand that students may be using different compilers. If your code does not compile properly during grading, we will check your source code manually only if it passes the following online compiler:

`https://www.onlinegdb.com/online_c++_compiler`

Therefore, students must ensure that their code compiles properly using the online test, otherwise they may receive 0 points.

Submission

- Please submit your source files through moodle. You should submit three source files (without compression) only, one for each task.
- Please name your files in format UniversityNumber-X.cpp for X in A, B, C, e.g. 1234554321-A.cpp.
- If you submit the files in the wrong format, marks will be deducted!