Algorithms and Data Structures 2022: First Practical Assignment

October 17, 2022

1 Instructions

The practical assignment consists of two parts, a program and a report.
You must write your program in Java, C++, C or Python 3. You can only use libraries that come with a minimal installation of the language. You must write all code yourself; copying code from fellow students or the internet is considered plagiarism.[1].
In your report, you must explain your algorithm, analyse the correctness and analyse the complexity. The report must have **at least two and at most ten** pages. Include your names on the first page of the report.

The deadline for the first practical assignment is **Friday, November 18th, 23:59**. You should submit your code and report via Brightspace. You can work either alone or as a pair (we encourage you to work in pairs, but the choice is up to you). Either way, you must enroll in a group for the first practical in brightspace. If you want to work in pairs, make sure you are enrolled in the same group. Only one team member has to submit the assignment.

2 Grading

Grades will be determined as follows. You may earn up to 100 points for your solution:

- 20 points for the explanation of your algorithm.
- 10 points for the correctness analysis.
- 10 points for the complexity analysis.
- 50 points for the test results. See section 4.1 for more (important) information.
- 10 points for the quality of the code.

The grade is the total number of points divided by 10. If you have questions, do not hesitate to contact Jelmer Firet, `jelmer.firet@ru.nl`.

[1] We will run a plagiarism checker. The results of this check are available without logging in. Therefore we do not require you to put your name or student number in your code.

3 The Problem: Restocking

Protesters have been blocking the distribution centre of a supermarket. Fortunately the night is falling and the protesters go home for sleep. The supermarket expects that the protesters will again block the supermarket tomorrow. This means there are only T hours to transport trucks with groceries to the supermarket.

The trucks use the road network to transport groceries from the distribution centre to the store. This road network consists of N cities and M highways between those cities. The distribution centre is in city 0 and the supermarket in city $N-1$.
When a truck enters any other city, they must leave immediately via a different highway. Trucks may leave the distribution centre from time $t = 0$ onwards, and should arrive at the supermarket at $t = T$ the latest (can also be earlier). You may assume there are plenty of trucks at the distribution centre and that they are all filled.

The road network is presented as follows:

- Highway i goes from city X_i to city Y_i, has length L_i and capacity C_i.
- The length L_i denotes the number of hours a truck takes to drive from X_i to Y_i.
- The capacity C_i gives us how many trucks can enter this highway at X_i every hour.

So, the problem you have to solve is: *what is the maximum number of trucks that can drive from the distribution centre to the supermarket within T hours?*

3.1 Input

Line 1		$N\ M\ T$	#locations, #highways and time available
Line $2+i$	for $0 \leq i < M$	$X_i\ Y_i\ L_i\ C_i$	from, to, length and capacity of highway i

3.2 Output

Line 1	S	Maximum number of trucks that can reach the supermarket in time.

3.3 Constraints

All variables are integers.
All testcases will satisfy the following constraints:

$0 < T \leq 50$ $\quad 2 \leq N \leq 50$ $\quad 0 \leq M \leq 1000$ $\quad 0 \leq S \leq 10^9$
$0 \leq X_i < Y_i < N$ $\quad 1 \leq L_i \leq T$ $\quad 1 \leq C_i \leq 10^7$ $\quad$ for $0 \leq i < M$

Most testcases (40 of the 50 points) satisfy $T \leq 30$, $N \leq 30$ and $M \leq 500$.
The other testcases are more challenging and require algorithms not taught in this course.

3.4 Examples

Sample input 1:

```
4 5 3
0 1 1 5
0 2 1 4
1 2 2 2
1 3 1 4
2 3 1 5
```

Sample output 1:

```
16
```

Sample input 2:

```
4 5 4
0 1 1 5
0 2 1 4
1 2 2 2
1 3 1 4
2 3 1 5
```

Sample output 2:

```
25
```

3.5 Explanation

The first and second testcase only differ by the time available to reach the supermarket. In both cases there are three routes the truck can take:

- $0 \to 1 \to 3$ takes 2 hours and has a capacity of 4 trucks.
- $0 \to 2 \to 3$ takes 2 hours and has a capacity of 4 trucks.
- $0 \to 1 \to 2 \to 3$ takes 4 hours and has a capacity of 2 trucks.

In testcase 1 we can send 4 trucks at hours 0 and 1 along each of the first two routes. The third route is too long to reach the supermarket in time. Therefore we can send $2 \cdot 4 + 2 \cdot 4 + 0 = 16$ trucks to the supermarket.

In testcase 2 we can send 4 trucks at hours $0, 1$ and 2 along each of the first two routes. Additionaly we can use the third route, but not to its full capacity. Because we already send 4 trucks each hour from 0 to 1 we can only send a single extra truck along the third route. Therefore we can send $3 \cdot 4 + 3 \cdot 4 + 1 = 25$ trucks to the supermarket.

We can not send more trucks in either of these cases.

4 Testing your code

You can test your program using DOMJudge (`https://domjudge.science.ru.nl`).
You should have received credentials from Jelmer via mail at the start of the course. You should ensure your program works in DOMJudge, as we will use it to judge your solution. During the last days before the deadline, the test server generally receives more submissions. Therefore the server will be slower in judging your program. To prevent this, you should start testing early.

You can also test your code locally. You can download the test cases from DOMJudge and feed them to your program. Redirect the `.in` file to the standard input of your program and compare the output with the `.ans` file.

```
./program <1.in >1.out && diff -bq 1.out 1.ans
```

4.1 Test results

We will run your program on a fresh set of test cases. The fresh set is generated in the same way as the set you can test your program with. You will get points for every test case where your program produces the correct answer within the time limit. If your code does not compile or does not read and write via `stdin` and `stdout`, you will get zero points on the test cases. Therefore, do test your code on DOMJudge! If you pass all test cases on the test server, you can assume you will get most of the points for the test results. If your program additionally finishes all test cases within a fraction of the time limit, you will likely get all points. Ask for help early if you encounter any issues!