Project 4 - Marathon

CmpE 250, Data Structures and Algorithms, Fall 2022

Instructor: Özlem Durmaz İncel
TAs: Suzan Ece Ada, Barış Yamansavaşılar
SAs: Batuhan Çelik, Bahadır Gezer, Zeynep Buse Aydın, Ömer Faruk Ünal

Due: 18/12/2022, 23:55 Strict

1 Introduction

A philosopher once said, "Some flowers... Some flowers don't grow in some soils. Such is life." A few days ago, I received such an e-mail from rambo.ocon@boun.edu.tr. He wrote in that e-mail, "As a traveling race man, I could not do anything when it came to coding. I'm not good enough to code and need help from cmpe250 students." So he asked me to create this project for you to help him. And now it's time to help him!

Esteban Ocon, a famous Formula 1 pilot, also known as Rambo Ocon, decides to participate in the 43rd Istanbul marathon, even though he has no previous running experience. Since he has a competitive personality, he wants to finish the race in first place. So he decides to cheat and complete the course in the shortest way possible.

In the meantime, he sees that at some points on the path, the opposing team's flags fly around. He wants to take down all these flags before the race starts. Because he doesn't want to get tired before the race, he wants to do it in the shortest possible way. Also, this will mean that the chances of him getting caught by the police will decrease; hence he will make it to the race.

Before the race, Rambo Ocon takes the map of Istanbul in front of him, marks the points where the flags are, and starts calculating the shortest route. Since there are many streets in Istanbul, and he is bad at math, he cannot calculate it. Besides, he doesn't know how to code, which is the icing on the cake. It's time to help Rambo!

2 Clarification

Let's say the race starts from Samandıra Tesisleri and finishes at Şükrü Saraçoğlu stadium. The flags are at 1st Istanbul Bridge and 2nd Istanbul Bridge.

Before the race, he must cut all the flags using the shortest possible path. There are no restrictions on the order in which he cuts the flags. He can start from the 1st Bridge, then go to the 2nd Bridge, or vice versa.

Thanks to public transportation, he can travel the same exact path as many times as he wants after traveling a path between two flags once, without getting tired. (More detailed explanation is given with the graphs in the More Examples section.)

During the race, he has to start from Samandıra Tesisleri and finish at Şükrü Saraçoğlu by running the shortest route.

- He has to cut all the flags, starting from any point with a flag.
- He can cut the flags in any order that will lead to the minimum distance path.
- There can be many different paths between any two flags.
- He can travel using the same point more than once. However, if he uses the same **exact** path between two flags, he can travel with public transportation (at no cost).
- **Exact** route: Let's say s1 and s4 contain flags. If Rambo Ocon uses the s1-s2-s3-s4 path without passing through any other points, then he can travel this same path with no cost (s1-s2-s3-s4 and s4-s3-s2-s1).
- He can use both directions of all point connections (e.g., graph is undirected).

3 Input & Output

3.1 Input

- The first line represents (V) the number of points in Istanbul (e.g., the number of all the nodes in the graph).

  $0 \leq E_i \leq 100$ *where* E_i *is the number of edge count for* V_i*.*
- The second line represents the total number of flags (M).

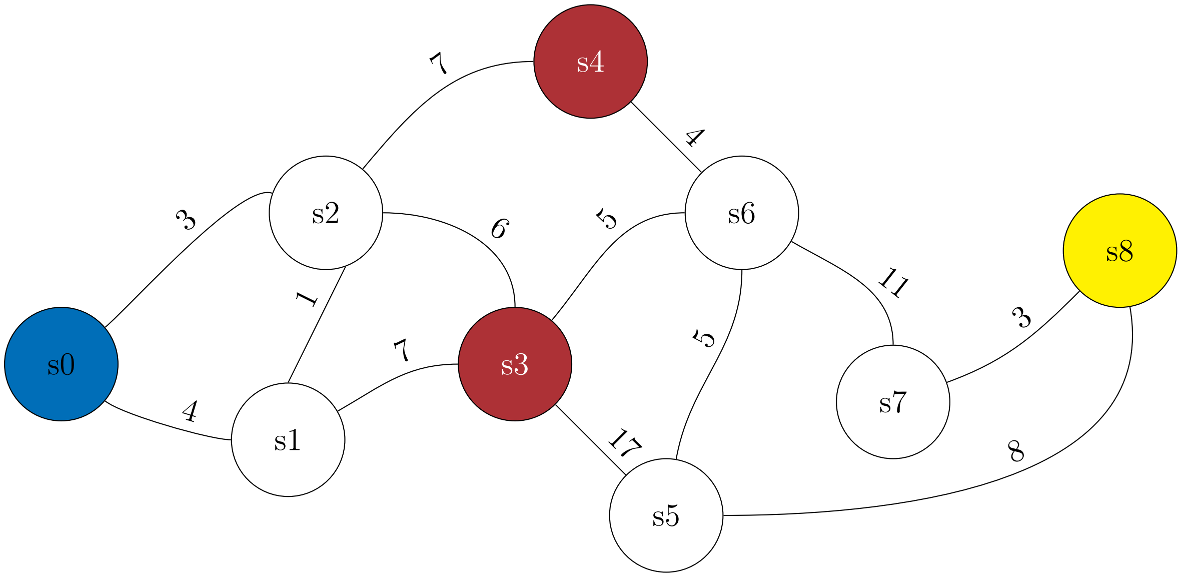
  $2 \leq V * M \leq 5.10^6$ and $2 \leq V \leq 5.10^6$
- The third line represents the name of the starting and ending points, respectively.
- The fourth line represents the name of the points where the flags are located.
- Each next line will give the name of a point in Istanbul and the names and lengths of the points that can be reached from that point in pairs. For example:

There are ways from s1 point to s2, s3, and s4 points of length 1, 2, and 3, respectively.

Then input line will be **s1 s2 1 s3 2 s4 3** If there is no way from s2, input line for s2 will be **s2**

Input	**Explanation**
9	The race path has 9 points.
2	There are 2 flags in the path.
s0 s8	The race will start from s0 point and end at s8 point.
s3 s4	The flags are located at s3 and s4
s0 s1 4 s2 3	The point with ID s0 has a way to s1 and s2 with length of 4 and 3 respectively.
	This also means that s1 has a way to s0 with length of 4.
s1 s2 1 s3 7	
s2 s3 6 s4 7	
s3 s6 5 s5 17	
s4 s6 4	
s5 s6 5 s8 8	
s6 s7 11	
s7 s8 3	
s8	

Table 1: Sample Input with Explanations



3.2 Output

- An integer, the length of the path during the race day. (If finishing the race is not possible, then there must be something wrong with the map so print -1.)
- An integer, the length of the path to cut the flags. (If cutting all the flags is not possible, then there must be something wrong with the map so print -1.)

Output	**Explanation**
27	The length (cost) of the path that Rambo used during the race. ($s0 \to s2 \to s4 \to s6 \to s5 \to s8$)
9	The length of the path that Rambo used to cut the flags. ($s3 \to s6 \to s4$)

Table 2: Expected Output File

4 Submission

Your code will be graded automatically. Therefore, it's important that you follow the submission instructions. This is your fourth project, so it is assumed that you are able to follow these instructions.

First, all of your source files should be collected under **Project4/src**. Then, you should zip the **Project4** folder and rename it to <**student**$_{id}$>**.zip**. This zip file will be submitted through moodle. Name of your main class **must** be **project4.java**.
Your program will be compiled with the command below:

```
javac -target 17 Project4/src/*.java -d ./
```

The input and output files can be in any folder. Design your code to accept the full path for file arguments. Your program will be run with the command below:

```
java project4 <inputfile> <outputfile>
```

Make sure your final submission compiles and runs with these commands.

5 Grading

Grading of this project is based on the automatic compilation, run, and the success of your code in test cases. If your code compiles and runs with no error, then you will get **10/100** of the project grade. The rest of your grade will be the sum of collected points from each test case.

You can also get partial credit for each input. If you find the length of the path Rambo used during the race, you will get **%30** from that test case and **%70** if you find the other length. The maximum grade for this project is 100. The **20** point worth of test case will be for checking the complexity value of your code with large input sizes.

6 Warnings

1. This is an individual project.
2. All source codes are checked automatically for similarity with other submissions and exercises from previous years. Do not copy codes from the internet or your friends. Make sure you write and submit your own code. Any sign of cheating will be penalized, and you will get **-50** points for the project, and you will get F grade in case of recurrence in any other project.
3. **There will be time limit on test cases, so the most important side of this project is to write your code in an efficient manner. Even if your code is completely correct, you will not get full points if it is not efficient.**
4. Parallel Programming is not allowed. The direct use of any parallel programming structure such as ForkJoin, Stream or Thread will result in a point reduction.
5. If the entire path you find is not suitable, you cannot get points for that route. We will not accept objections like: ”but my route is correct except for the last two points.”
6. You can add as many files as you can as long as they are in the “src” folder and your project can be compiled as above. But the entry point of your program should be “project4.java”.
7. Make sure you document your code with necessary inline comments and use meaningful variable names. Do not over-comment or make your variable names unnecessarily long. This is very important for partial grading.
8. Expected running time of each test case is provided with input/output files. They are not strict limits; it is given so that you have an idea about the approximate running times. Execution times may vary for each run or computer to computer. If it works in 10 seconds on my computer, it can run in 20 seconds on another computer. However, if it’s running in 100 seconds on another computer, there might be something wrong with your code.

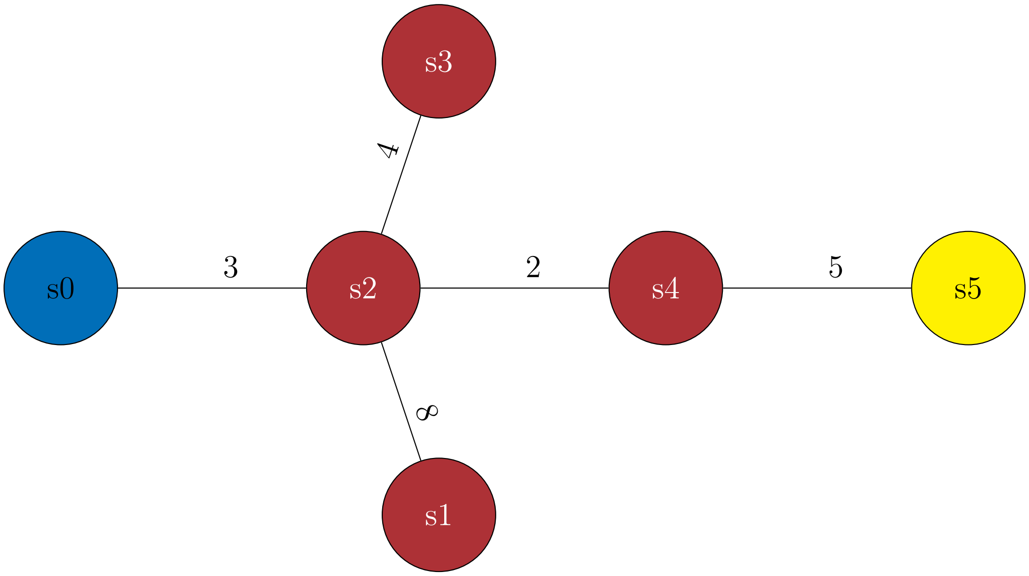
7 More Input & Output Examples

7.1 Example1

Input
6
4
s0 s5
s1 s2 s3 s4
s0 s2 3
s1 s2 8
s2 s3 4 s4 2
s3
s4 s5 5
s5

Output	Explanation
10	$(s0 \rightarrow s2 \rightarrow s4 \rightarrow s5)$
14	$(s1 \rightarrow s2 \rightarrow s3 \rightarrow s2 \rightarrow s4)$

Table 3: Sample Input2 and Output2 with Explanations



Explanation: While cutting the flags, Rambo starts from s1. Then he goes to s2. Then he goes to s3. He traveled 12 distances in total. Since he used **exactly the same path** between s2-s3 before, he uses public transport here and returns to s2 again. His total distance is still 12. Then he goes to s4 and finishes cutting the flags with a distance of 14.

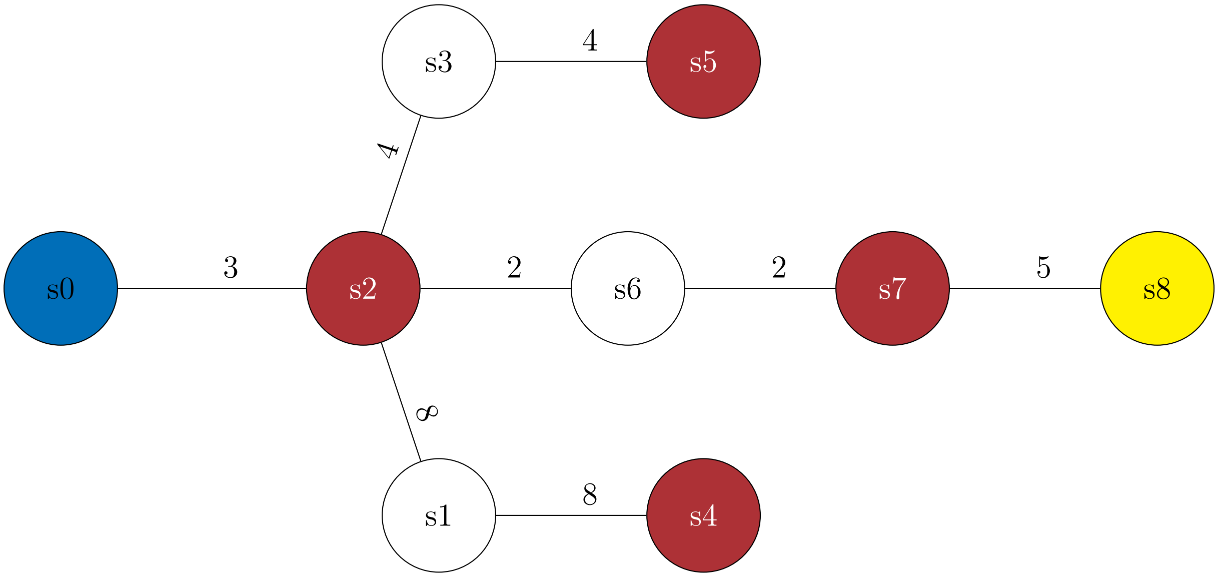
Note that it is also okay to start from s2, s3, or s4. All of them will give the same result. But he can't start from s0 or s5.

7.2 Example2

Input
9
4
s0 s8
s2 s4 s5 s7
s0 s2 3
s1 s2 8 s4 8
s2 s3 4 s6 2
s3 s5 4
s4
s5
s6 s7 2
s7 s8 5
s8

Output	Explanation
12	$(s0 \rightarrow s2 \rightarrow s6 \rightarrow s7 \rightarrow s8)$
28	$(s4 \rightarrow s1 \rightarrow s2 \rightarrow s3 \rightarrow s5 \rightarrow s3 \rightarrow s2 \rightarrow s6 \rightarrow s7)$

Table 4: Sample Input3 and Output3 with Explanations



Explanation: While cutting the flags, Rambo starts from s4 and follows the path s1, s2, s3, s5 and the total distance is 24. Because he used the exact s5-s3-s2 path before, he goes to s2 at no cost. Then he goes to s6 and s7. The spanned total distance is 28.

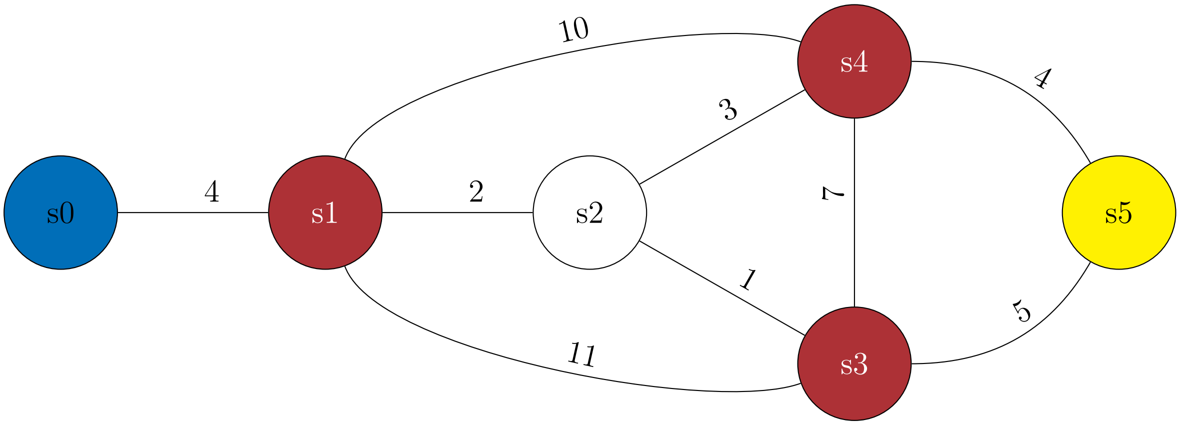
Note that it is also okay to start from s7, s5 or s2. All of them will give the same result. However, he can't start from any other nodes without flags.

7.3 Example3

Input
6
3
s0 s5
s1 s3 s4
s0 s1 4
s1 s2 2 s3 11 s4 10
s2 s3 1 s4 3
s3 s4 7 s5 5
s4 s5 4
s5

Output	Explanation
12	$(s0 \rightarrow s1 \rightarrow s2 \rightarrow s3 \rightarrow s5)$
7	$(s1 \rightarrow s2 \rightarrow s3 \rightarrow s2 \rightarrow s4)$

Table 5: Sample Input4 and Output4 with Explanations



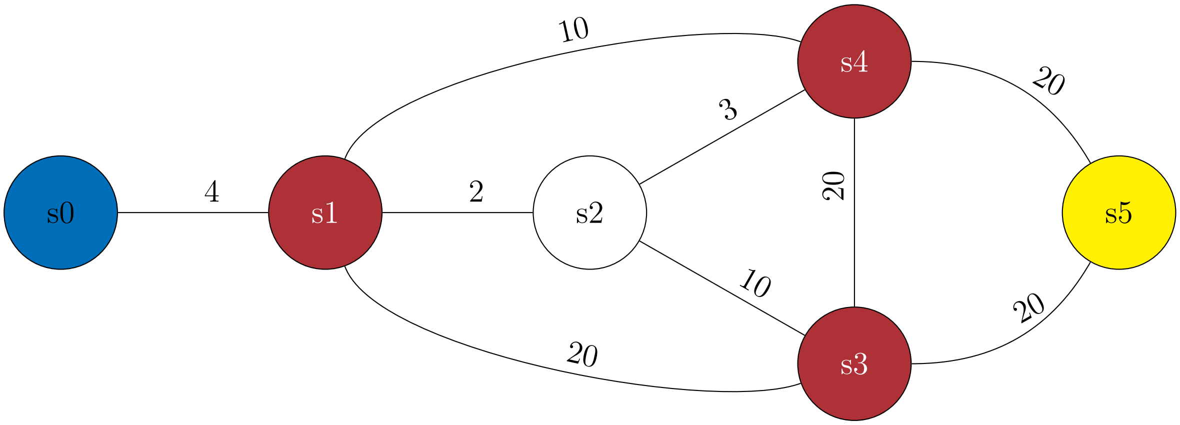
Explanation: While cutting the flags, Rambo starts from s1. Then follows the path s2-s3 with a total distance of 3. Now he has to go to s4 but because he never traveled the **exact path** s3-s2-s4, he can't use public transportation. He has to follow s2, s4 with a total distance of 3+4 = 7.

7.4 Example4

Input
6
3
s0 s5
s1 s3 s4
s0 s1 4
s1 s2 2 s3 20 s4 10
s2 s3 10 s4 3
s3 s4 20 s5 20
s4 s5 20
s5

Output	**Explanation**
29	$(s0 \rightarrow s1 \rightarrow s2 \rightarrow s4 \rightarrow s5)$
17	$(s1 \rightarrow s2 \rightarrow s3 \rightarrow s1 \rightarrow s2 \rightarrow s4)$

Table 6: Sample Input4 and Output4 with Explanations


10
s4
20
3
20
s0
4
s1
2
s2
s5
10
20
20
s3