Comp 305 Spring 2021 — Algorithms and Complexity — Final Project Group 16

1 Introduction

In this project, you are given a programming question, and you will develop an efficient algorithm and implement it. You can use any of the programming languages you prefer, there is no limitation. However, usage of any libraries except the standard language library are forbidden. Please request permission if you plan to use any of these (if you are using Python, NumPy is allowed).

Please open the Google Sheet Link, and enter your group number, names of the members and the emails of the members to the cell, where your project is listed. Please keep in mind that the sheet document is divided into different sub-sheets. Thus, select the correct sub-sheet with respect to your group number. All of the test cases for each of the projects can be viewed and downloaded from this Google Drive Link.

Please open a GitHub public repository, include all of your members as contributors and add the repository link to the given Google Sheet document. This step is quite important for us to see your progress and has to be done quickly. Therefore, in the README file please keep a list of completed steps, a TO DO list and the results retrieved if there are any. You will also prepare a presentation and present to the TAs. Therefore, you should also create a Google Slides presentation and include its link to the Google Sheet document.

Please email to *comp305staff-group@ku.edu.tr*, if there is any problem in viewing the drive folder or modifying the document or if you have some troubles with any of the test cases.

2 Presentation Details

Each of the presentations should take ∼10 minutes and there will be a 5-minute Q&A session afterwards. If a presentation lasts longer than 10 minutes, then it will be interrupted. During the presentation each of the groups should explain and report:

- The algorithm you designed to solve the problem, the choices of the data structures you used and your reasoning.
- The time complexity of your algorithm (and the space complexity if applicable).
- Your run times for each of the test cases.
- Further improvements that can be done as future works.

This project does not expect from you to come up with just one solution and then test only that solution. For each of the problems you can start with some baseline approaches with more complexity and improve the baseline algorithm step by step. Be as creative as possible. Report different approaches you tested and why did you decide on the final algorithm you present. Your grading will be based on your creativity, your cumulative progress and how well did you present your approach. Additionally, there will be more test cases which won't be provided to you. Therefore, to be sure about the correctness of your algorithm, you should create extra test cases. Finally, having an efficient and fast algorithm is also as important as having correct algorithm. Do your best to find the most efficient and fastest solution. Your effort is really important for grading.

3 Deadlines

You can work on your project until the end of ***23th May, 2021***. The project presentations will be held between ***24th-28th of May, 2021***.

In the following pages, you can see the available project(s):

Gezgin and Bezgin

Gezgin and Bezgin are 2 friends who are aiming to travel Europe, however because of the inflation in their country's economy, they should be careful while spending their money. Also they need to decide on the number of country they want to visit before the travel and according to their goal they will create their plan. While they are planning their travel, they examined several approach. Firstly, they thought that they can use use plane while traveling but they quit that idea since flight tickets are expensive. Then Gezgin proposed to use newly created mobile app in which you can be guest the other's traveler's car and routes of the this cars are predefined by the driver. In addition to that, the places that you get into the car is also predefined. In other words, start and end points of the route is predefined by the driver. In the mobile app, you have 2 options to buy ticket. First option is you can buy only one way ticket which means you can only travel with 1 driver. Second option is you can pay for MAXI-TICKET in which you can use it on up to t number of consecutive travels in one day. For instance if you buy 4 way tickets, you can use it to travel from Points A to Point B then from Point B to Point C (2 Tickets are used up to now) and from Point C to point D (3 tickets are used up to now) and from Point D to Point E (4 tickets are used and maximum number of travel reached). The software engineers of the app decided to develop new feature such that, at each point according to ticket type number of possible starting points will be shown to user.

In the figure there are cities that they want to visit. Assume that they both have 2 credits tickets. Arch implies that there is a car in the app which goes from starting position to end position. For instance there is transportation between Amsterdam and Paris in the app. Red cities implies that possible starting points if you have MAXI TICkET with 2 credits. For example, there is no way to come to Amsterdam and therefore you can only start at Amsterdam and that is why there is only red Amsterdam on the top of Amsterdam circle. For Berlin, since you have 2 credits tickets, they can start at Amsterdam and move Paris and consume 1 ticket and from Paris they can arrive Berlin by consuming 2 tickets so Amsterdam as starting point is possible. In addition to that, they can start at the Paris and consume 1 ticket and arrive in Berlin so Paris is possible. Hamburg is also possible starting point with the same reasoning. For Vienna however, there is no read Amsterdam on top of it since with 2 tickets it is not possible to start at Amsterdam and arrive at Vienna.

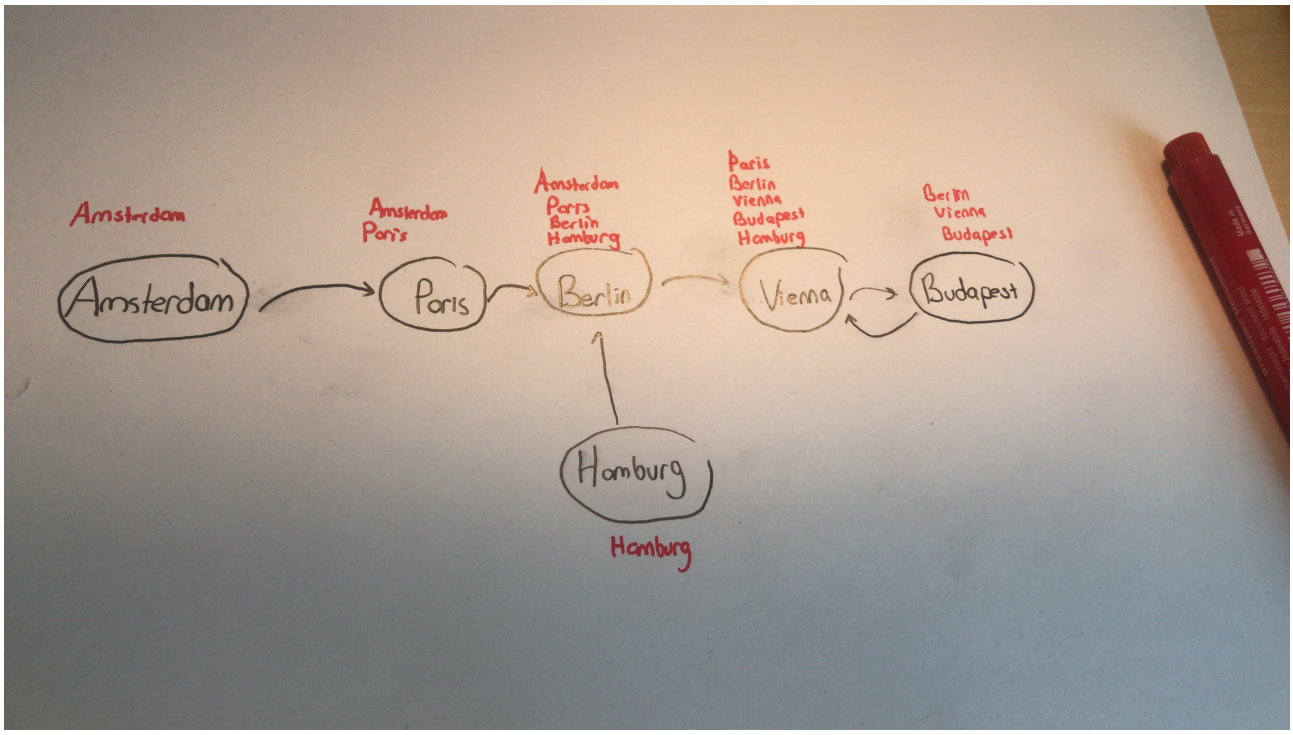

Figure 1: Archs represents there is available transportation from one city to another. Red writings resemble possible starting positions when you buy MAXI TICKET with 2 credits.

4.1 Inputs

The first line in the input is number of cities that they want to travel and credits that have in their MULTI-CARD. Then the following lines lines are the integer which resembles a city that can be reached from the that index's city with 1 tickets. For instance, lets say i^th line has a integer 20, that means that there is a available transportation from i^th to 20th city with one ticket. To be more clear, in the upper figure lets say Amsterdam is the city 5 and Paris is the city 6. 5th line in the input file is 6.

4.2 Output

There will be number of cities that they want to visit -n- lines in total output file. And i^th line contains the number of possible starting point to reach to i^th with the credits that they have in their MULTI-CARD. In other words, for the upper figure, Paris has 2 possible starting positions namely Amsterdam and Paris so line that containing Paris should be 2.

4.3 Sample Input Output

Each city in the upper figure is represented with unique integer.

Sample Input 1

```
6 2
2
3
4
5
4
3
```

Sample Output 1

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

Sample Input 2

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

Sample Output 2

```
3
3
3
2
2
```

Figure 2: Sample Input and Output File.

Question Given these information, your task is find for each target city that are planned to visit, find all the possible starting points given that they have MULTI TICKET with k credits. Provide the most efficent algortithm that you can since your algorithm will be tested with big input size.