Comp 305 Spring 2021 — Algorithms and Complexity — Final Project Room 21

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.

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 assigned project:

Pokemon Center Planning for the Upcoming Tournement

There is big pokemon tournement coming. All over the world trainers are competing with each other. However, during the competitions trainers and pokemons might get hurt. If that happens they should immediately find a nearby Pokemon Center and see Nurse Joy. However, there can not be a Pokemon Center in each town.

As a result, only some cities can have centers. The challenge with doing this is to figure out where to put centers so that (1) we don't spend too much money for the expenses more than we need, and (2) we don't leave any cities too far away from Pokemon Centers. Namely, cities either should have a PokeCenter or needs to have direct access (neighbor city) to a city that has one.

(a) We'll say that a country is tournement-ready if every city either already has Pokemon Centers or is immediately down the highway from a city that has them. Your task is to write a program

```
algorithm canBeMadeTournementReady is
    input: set of roads as roadNetwork, number of cities(numCities)
    output: set of pokeCenters as set of strings
```

that takes as input a HashMap representing the road network for a region (described below) and the number of cities you can afford to put PokeCenters in, then returns whether it's possible to make the region tournement-ready without placing PokeCenters in more than numCities cities. If that is the case, the program should then populate pokeCenters to be returned with all of the cities where PokeCenters should be stored.

In this problem, the road(town) network is represented as a map. In this map, a city is represented with a key and set of cities are the values which are immediately down the road from them.

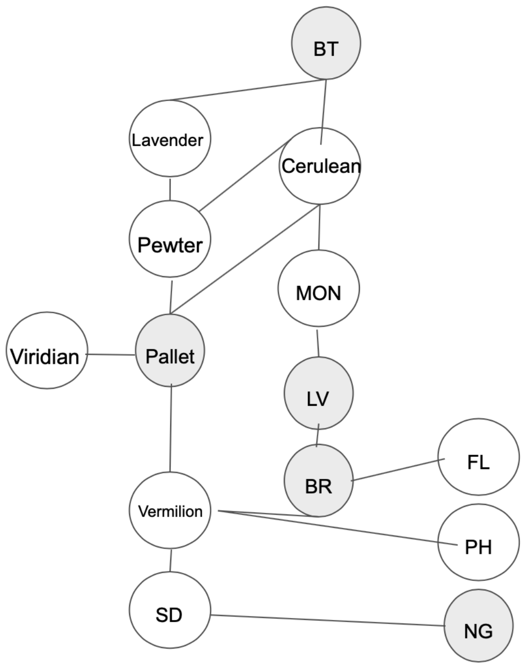

Figure 1: Example Map

Example Test Case You can see that some parts of the above town network can be shown as follows:

- "Pallet Town": "Viridian City", "Pewter City", "Cerulean City", "Vermilion City"
- "Viridian City": "Pallet Town"
- "Pewter City": "Lavender Town", "Pallet Town", "Cerulean City"

Important Points

- The network of roads is bidirectional meaning that if there is a road between town A to B it means there is also way from B to A.
- Every town appears as a key in the map. There might be cities which are not adjacent to any other cities. In that case the value for that key would be empty set.
- The numCities parameter denotes the maximum number of cities you're allowed to PokeCenters in. It's absolutely fine if you can manage to use less than numCities cities to cover all the map, but you can't use more.
- The numCities parameter may be zero, but should not be negative. If it is negative, raise error.