Comp 305 Spring 2021 — Algorithms and Complexity — Final Project Room 22

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:

Tribal Shaman King Selection

The country of Triberia is a forgotten Shamanic civilization that lived approximately around 20,000 BC. They were a very progressive society in the sense that they were selecting their shaman king. However, The Shaman King of the Triberia is not elected by a popular vote, but by a majority vote in the Electoral Hut. Each tribe in Triberia, gets some number of king-selectors in the Electoral Hut, and whoever they vote in becomes the next Shaman King.

Assumptions

- You need to win a majority of the votes in a tribe to earn its king-selectors, and you get all the tribes's king-selectors if you win the majority of the votes. For example, in a small tribe with 999,999 people, you'd need 500,000 votes to win all its king-selectors. These assumptions aren't entirely accurate, both because in most tribes a plurality suffices and some tribes split their electoral votes in other ways.
- You need to win a majority of the electoral votes to become Shaman King. In the 20,008 BC election, you'd need 270 votes because there were 538 electors. In the 18,004 BC election, you'd need 89 votes because there were only 176 electors.
- King-Selectors never defect. The King-Selectors in the Electoral Hut are free to vote for whomever they please, but the expectation is that they'll vote for the candidate that won their home tribe. As a simplifying assumption, we'll just pretend King-Selectors always vote with the majority of their state.

(a) under these assumptions, what's the fewest number of popular votes you can get and still be elected Shaman King?. Your task is to write a program

```
algorithm select-shaman-king-with-min-votes is
    input: list of tribes
    output: number of min required votes
```

that takes as input a list of all the tribes that participated in the election, then returns some information about the minimum number of popular votes you'd need in order to win the election (namely, how many votes you'd need, and which tribes you'd carry in the process). Here's a quick overview of the types involved here. First, there's the Tribe type, defined here:

```
struct Tribe {
string name; // The name of the tribe
int electoralVotes; // How many selectors it has
int populationVotes; // The number of people in that tribe who voted
};
```

The input to select-shaman-king-with-min-votes is a List¡Tribe¿ containing information about all the tribes that participated in the election. The select-shaman-king-with-min-votes function then returns a min-required-vote-info, a type that contains information about the minimum popular vote needed and which tribes you'd carry:

```
struct MinRequiredVoteInfo {
int populationVotesNeeded; // How many commoner (people that vote for king-selectors) votes you'd need
};
```

Example Test Case

20,016 BC Shaman King Selectoral Attendance

- 20,016BC (Selection Year) ,29231179 (total votes) , 8221109 (populationVotes needed)
- Chatan, 9.(King Selector Number) ,2123372 (Population of this tribe)
- Otoahnacto, 3,318608
- Shappa,11,2604657
- Ocunnowhurst,6,1130635

- Chuslum,55,14237884
- Tokala,9,2780247
- Kangee,7,1644920
- Matunaaga,3,443814
- Honovi,3,311268
- Neeheeoeewootis,29,9501617
- Harkahome,16,4141445
- Alahmoot,4,428937
- Cameahwait,4,690433
- Nimeda,20,5589767
- Honiahaka,11,2757828
- Chosovi,6,1566031
- Atepa,6,1194755
- Kokyangwuti,8,1924149
- Tsomah,8,2029032
- Taigi,4,747927
- Sinopa,10,2781446
- Shadi,11,3325046
- Tadewi,16,4824119
- Mammedaty,10,2945233
- Nashashuk,6,1211088
- Tapco,10,2827673
- Otaktay,3,501822
- Satanta,5,844227
- Ahuli,6,1125385
- Keme,4,744296
- Ahote,14,3906723
- Osceola,5,798318
- Enapay,29,7707363
- Mahu,15,4741564
- Bonita,3,344360
- Kamali,18,5536528
- Nokomis,7,1452992
- Tangakwunu,7,2001336
- Chosovi,20,6166698

- Tansy,4,464144
- Kuwanyauma,9,2103027
- Asdza,3,370093
- Nidawi,11,2508027