Audie's party

Algoritmia II

March 14, 2024

At the beginning of the year, Audie moved to a new town. These last few weeks, she has taken it upon herself to meet the villagers of the new place where she lives.

She decided to throw a big party to bond more with her neighbors. She knows that some villagers get along better with some than others, so even though she would like to invite everybody to her party, she will have to decide whom to invite to have a successful party.

Audie got a Txt file with the villagers' names, followed by a list of two names and a numerical value representing how well they get along. (Their friendly relationship, the bigger the value, the better they get along)

Audie wants to create her guest list based on how well the villagers get along. Help Audie to:

Generate a list where all the villagers that are part of it are the ones that get along with a value greater than **x** with any other villager on the list.

- Audie wants to invite as many guests as possible, so the list should be the biggest.

- If several answers are found, print the list whose sum of their friendly relationship among all the guests is the greatest. If there is still a tie, print any.

After having the list of guests she should invite to her party, she wants to know if dividing them into exactly **k** groups is possible so the villagers that get along very well are together.

- She wants to know the names of the villagers in each of the groups that will be formed. (A villager can be part of only 1 group)

  - Print the group with the guest with the strongest friendly relationship. (If it does not exist print "None")

  - Print the group with the guest with the least friendly relationship. (If it does not exist print "None")

- If multiple answers, print any of them.

- If it is not possible, print "It is not possible"

E.g. 1

Txt file with the list of Audies' neighbors and their connections:

```
Judy
Kitt
Cube
X
M
P
Judy X 10
Cube Judy 12
Kitt X 7
X M 2
P Kitt 20
Cube P 5
Judy Kitt 13
```

Case 1

Input:
x = 7
Output:
Judy Kitt Cube X P

Input:
k = 2
Output:
P Kitt Judy Cube
X
Group with strongest friendly relationship: P Kitt Judy Cube
Group with least friendly relationship: P Kitt Judy Cube

E.g. 2

Txt file with the list of Audies' neighbors and their connections:

```
A
B
C
X
R
M
N
A X 12
X N 8
X R 11
R M 20
R C 25
A B 15
N R 13
```

Case 1

    Input: x = 10
    Output:
        A B C X R M N

    Input: k = 4
    Output:
        C R M
        A B
        X
        N
        Group with strongest friendly relationship: C R M
        Group with least friendly relationship: A B

Case 2

    Input: x = 10
    Output:
        A B C X R M N

    Input: k = 3
    Output:
        C R M N
        A B
        X
        Group with strongest friendly relationship: C R M N
        Group with least friendly relationship: C R M N

Case 3

    Input: x = 14
    Output:
        C R M

    Input: k = 3
    Output:
        C
        R
        M
        Group with strongest friendly relationship: None
        Group with least friendly relationship: None

Instructions

1. Implement the necessary code to solve the problem.

   (a) Make sure that you apply **greedy** algorithms to your solution.

   (b) Do not forget to write clean code and follow best practices.

2. Write a brief explanation of why you have chosen the greedy algorithm to solve the problem.

   (a) If you use a new algorithm, include the correctness proof; otherwise, you only need to mention the algorithm's name.

3. Identify the time complexity of your solution.

Submission

- For the code, you can either submit it to assignments in Teams or push it to your private repository, add my user, and give me the necessary permissions to view it

- For the other points, you should submit your answers in a PDF or have the answers in the README of your repository.

This is an individual assignment