# Introduction about Programming Assignment – Group 4

## 1 Introduction

521 programming assignment for our group 4 is:

p743 question 20

***Given the distances between pairs of television stations and the minimum allowable distance between stations, assign frequencies to these stations.***

Based on our understanding, this is an undirected graph coloring question with a minimum distance additional condition.

**This mean that:**

1. **For two stations that have the distance shorter than minimum allowable distance definitely have different frequencies. Moreover, we can set an edge between these two stations.**
2. **For two stations that have longer distance than minimum distance, we don’t add edge between these stations.**
3. **After 1) and 2) we can get a graph then it becomes a graph coloring questions for the frequencies assignment.**

For this assignment there are just two main jobs:

1. Create the graph and the adjacencies’ relation of each station by input data, the graph is undirected graph.
2. Find an algorithm to solve the coloring question.

The rest of this document will show you our algorithm for solving the problem (with two examples) and program flow chart for coloring process.

## 2 Project and Group work participation Introduction

We use java to solve the question. You can check the project in Eclipse Version Neon or Oxygen Release (4.7.0).

The zip file includes all of the related files as project, document and output snapshot.

We have also created a **github repository** for group work, the link is:

[***https://github.com/521Group4/Group4ProgrammingAssignment***](https://github.com/521Group4/Group4ProgrammingAssignment)

Group work participation shows below:

|  |  |
| --- | --- |
| Work | Involved members |
| Algorithm providing | All members |
| Coding | Yunwei Jiang, Karyn Doke |
| Testing | All members |
| Document writing | Yunwei Jiang, Jingzhou Hu |

## 3 A Simple example

**We can use a simple example to illustrate this assignment.**

For three stations like below, suppose the minimum allowable distance is **4**:

**2**

**3**

**3**

**3**

**1**

**5**

1. We can get a distance relation form as below:

|  |  |
| --- | --- |
| Stations | Distance relations |
| 1 | {{Station1,0}, {Station2,3}, {Station3,5}} |
| 2 | {{Station1,3}, {Station2,0}, {Station3,3}} |
| 3 | {{Station1,5}, {Station2,3}, {Station3,0}} |

1. Then this form can be represented as a 3X3 Symmetric Matrix:

|  |  |  |  |
| --- | --- | --- | --- |
| Station | 1 | 2 | 3 |
| 1 | 0 | 3 | 5 |
| 2 | 3 | 0 | 3 |
| 3 | 5 | 3 | 0 |

1. Adding edges between vertexes

Check the Matrix, if the value > 4, then add edge between the vertexes. The vertexes’ adjacencies form is as follow:

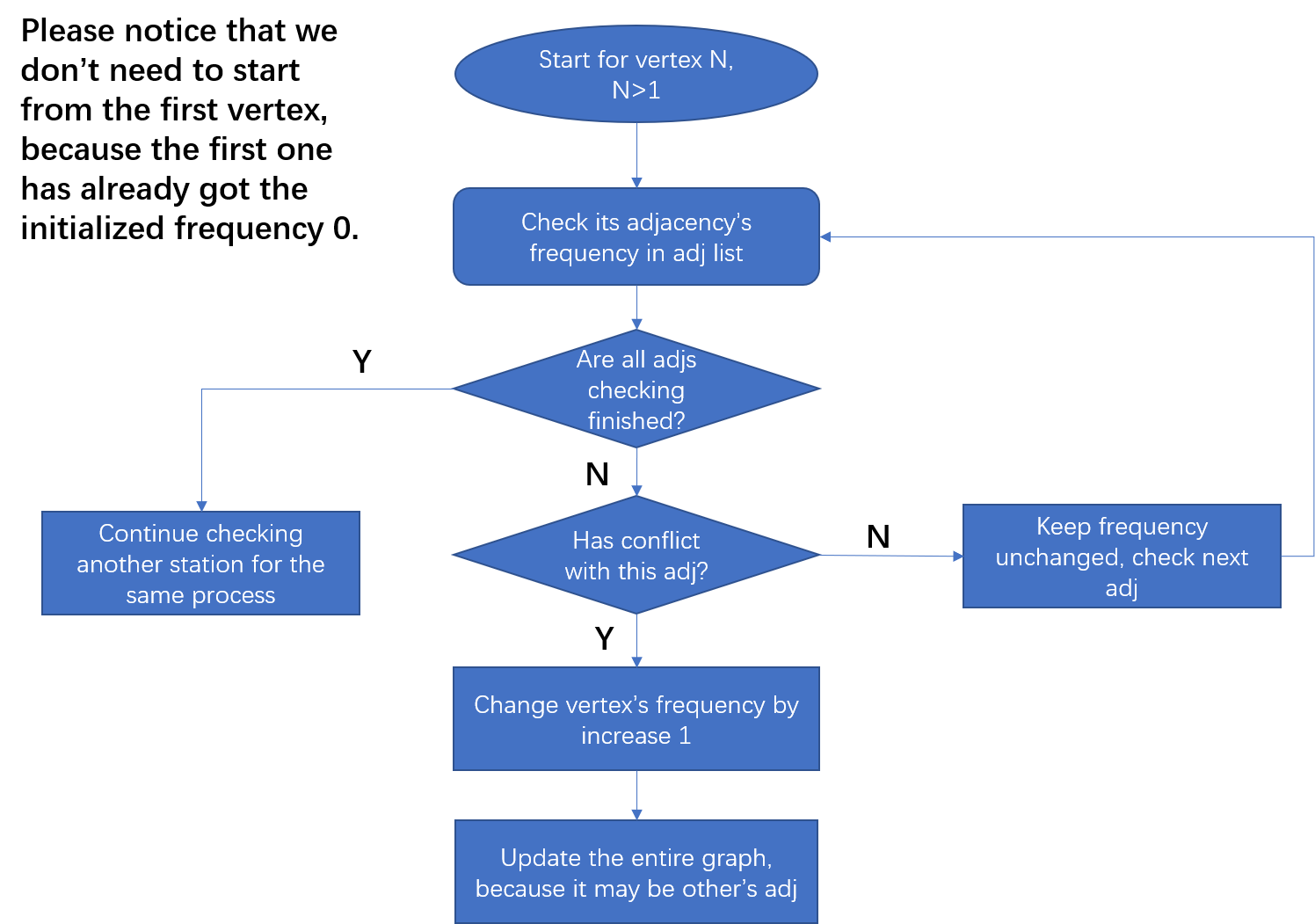
|  |  |
| --- | --- |
| Stations | Adjacencies relation |
| 1 | { Station2 } |
| 2 | { Station1, Station3 } |
| 3 | { Station2 } |

1. Coloring the three stations

The coloring process for each vertex or station in the graph shows below:

For initialization the origin frequency for each station is 0.

**Also need to notice that we don’t need to start from the first vertex, because the first one has already got the initialized frequency 0.**



|  |  |
| --- | --- |
| Iteration | Frequency result |
| 1 | {{Station1, 0}, {Station2, 1}, {Station3, 0}} |
| 2 | **{{Station1, 0}, {Station2, 1}, {Station3, 0}}** |

After two iterations, we get the result: {{Station1, frequency0}, {Station2, frequency 1}, {Station3, frequency 0}}.

## 4 A slightly complicated example

Suppose there are 8 stations (from station 0 to 7) with the distances showed below:

**Note: This example is what we solved in source code.**

Matrix of distance

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Station | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 2 | 5 | 8 | 7 | 6 | 5 | 9 |
| 1 | 2 | 0 | 9 | 7 | 8 | 4 | 3 | 2 |
| 2 | 5 | 9 | 0 | 5 | 6 | 8 | 9 | 5 |
| 3 | 8 | 7 | 5 | 0 | 4 | 2 | 3 | 9 |
| 4 | 7 | 8 | 6 | 4 | 0 | 8 | 7 | 5 |
| 5 | 6 | 4 | 8 | 2 | 8 | 0 | 2 | 5 |
| 6 | 5 | 3 | 9 | 3 | 7 | 2 | 0 | 4 |
| 7 | 9 | 2 | 5 | 9 | 5 | 5 | 4 | 0 |

Each row means the distance between other stations, and this is a symmetric matrix.

Suppose the minimum distance is **4. So for distance shorter than 4 we add an edge.**

### Step 2 Adding edges based on Matrix

The edge/adjacencies relation for each station shows below:

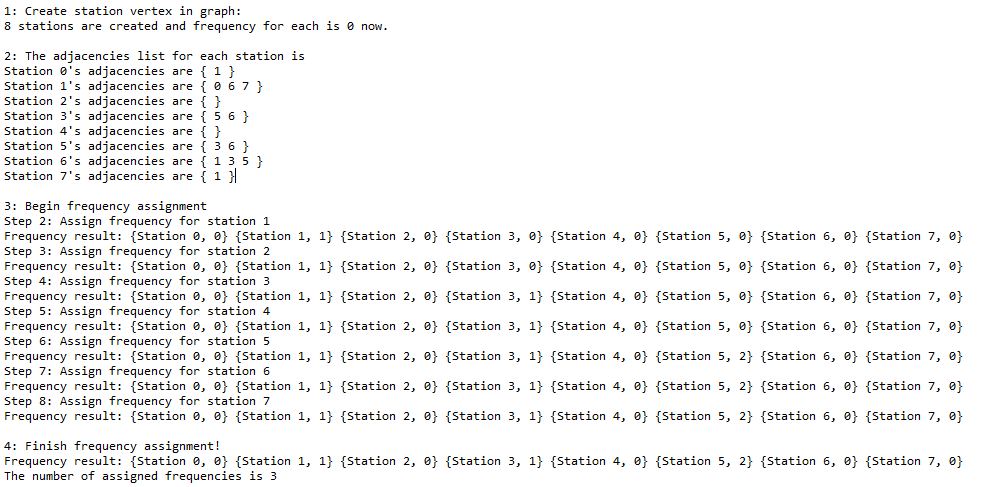
|  |  |
| --- | --- |
| Station | Edge relation/ Adjacency stations |
| 0 | 1 |
| 1 | 0 6 7 |
| 2 | none |
| 3 | 5 6 |
| 4 | none |
| 5 | 3 6 |
| 6 | 1 3 5 |
| 7 | 1 |

### Step 3 Adding edges based on Matrix

The rest of the work is to do the same process showed in the former simple example.

For 8 stations, we need 7 times iteration to finishing the coloring process. The result for each iteration shows below:

The result shows below:



So now we got the frequency List for these 8 stations are

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| station | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Color value | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 |