# 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 a 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, just leave their frequencies unchanged.**

## 2 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. Using the minimum distance 4 to change this Matrix to a 0/1 Matrix (if distance > 4 then set to 1, else 0):

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

1. Coloring the three stations

For this assignment there are just two main jobs:

1. Create the graph class and create the graph by input data, the graph is undirected graph.
2. Find an algorithm to solve the coloring question.

## 2 A slightly complicated example

Suppose there are 8 stations with the distances showed below:

Matrix of distance

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

### Step 1 Change this matrix to a 0/1 matrix by using the minimum distance 3

If the distance between two stations equals to or shorter than 3, set the distance to 0; if not, set to 1.

So we can get the new matrix:

0/1 Matrix of distance

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

Also set a list of color for 8 stations, all just initialized to color 0, **colorLists** = {0,0,0,0,0,0,0,0,}

### Step 2 Coloring the stations in the matrix – for the 1st row

Using the following step to coloring the stations: D(i,j) for Distance between station i and j, C(j) is the color of station j; color for recording how many color used now.

Start from station 1 and set its color to **0, C(1) = 0; color = 0;**

**The algorithm for coloring could be:**

**int counter = 0;**

**For all of the station I, 1<=i<=8{**

**For all of the stations j, 1<=j<=8{**

**counter = 0; // To record if how many times that D(i,j) == 1 && C(i) == C(j) && j != i in one inner for loop //cycle**

**If D(i, j) == 0 {**

**//This means I and j are shorter than minimum station distance**

**// if the color of I and j are different, just set C(j) the same to C(i)**

If(C(i) != C(j))

**C(j) = C(i);**

**}**

Else if D(i, j) == 1{

//if they have the same color and j != i, then station j changed the color according to **two situations**

If(C(i) == C(j) && j != i){

**counter++;**

**//if this is the first time that D(i,j) == 1 && C(i) == C(j) && j != i**

**if(counter == 1)**

**{**

//分情况考虑，若在这次循环中这是第一次出现则C(j) = findMaxColor(colorLists) + 1；

**//** findMaxColor(colorLists) means find the biggest color value in colorLists for 8 stations

C(j) = findMaxColor(colorLists) + 1；

**}**

**Else**

**{**

//若非第一次出现则C(j) = findMaxColor(colorLists);

**//if not first time, just set C(j) to the return of findMaxColor**

C(j) = findMaxColor(colorLists);

**}**

}

} //end of Else if D(i, j) == 1

}//end of inner for loop

**}//end of out for loop**

**Using the algorithm for the first row in matrix:**

**When i = 1** after first cycle, the Matrix is changed to:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Station | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 C(1)=0 | 0  C(2)=0 | 1  C(3)=1 | 1  C(4)=1 | 1  C(5)=1 | 1  C(6)=1 | 1  C(7)=1 | 1  C(8)=1 |
| 2 C(2)=0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 5 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

**Now the color list is {0,0,1,1,1,1,1,1}**

### Step 3 Coloring the stations in the matrix – for the rest rows(2~7)

For the second cycle, when i = 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Station | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0  C(1)=0 | 0  C(2)=0 | 1  C(3)=1 | 1  C(4)=1 | 1  C(5)=1 | 1  C(6)=1 | 1  C(7)=1 | 1  C(8)=1 |
| 2 C(2)=0 | 0  C(1)=C(2) | 0  C(2)=C(2) | 1  C(3)=C(3) | 1  C(4)=C(4) | 1  C(5)=C(5) | 1  C(6)=C(6) | 0  C(7)=C(2) | 0  C(8)=C(2) |
| 3 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 5 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

**Now the color list is {0,0,1,1,1,1,0,0}**

For the 3rd cycle, when i = 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Station | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0  C(1)=0 | 0  C(2)=0 | 1  C(3)=1 | 1  C(4)=1 | 1  C(5)=1 | 1  C(6)=1 | 1  C(7)=1 | 1  C(8)=1 |
| 2 C(2)=0 | 0  C(1)=0 | 0  C(2)=0 | 1  C(3)=1 | 1  C(4)=1 | 1  C(5)=1 | 1  C(6)=1 | 0  C(7)=C(2)=0 | 0  C(8)=C(2)=0 |
| 3 C(3)=1 | 1  C(1)=0 | 1  C(2)=0 | 0  C(3)=1 | 1  C(4)  =findMaxColor+1  =1+1=2 | 1  C(5)  =Maxcolor  =2 | 1  C(5)  =Maxcolor  =2 | 1  C(7)=0 | 1  C(8)=0 |
| 4 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 5 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

**Now the color list is {0,0,1,2,2,2,0,0}**

Just continue doing this for the rest 5 cycles

|  |  |
| --- | --- |
| Cycle time | colorList |
| 4 | 0 0 1 2 3 2 2 0 |
| 5 | 0 0 1 2 3 2 2 0 |
| 6 | 0 0 1 2 3 2 2 0 |
| 7 | 0 2 1 2 3 2 2 0 |
| 8 | 4 0 1 2 3 2 2 0 |

So now we got the colorList for these 8 stations are

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| station | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Color value | 4 | 0 | 1 | 2 | 3 | 2 | 2 | 0 |

You can actually check if this coloring is right using the 0/1 distance matrix, for example check station 3:

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

The color of Stations 3 is 1, and is different to station 1,2,4,5,6,7,8’s colors