ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 336E

ANALYSIS OF ALGORITHMS II
PROJECT II REPORT

STUDENT NAME : Fatma Sena Akçağlayan

STUDENT ID : 150170085

ASSINGMENT DATE : 22.04.2022

SPRING 2021-2022

Contents

1 Question 1

Regions	1	2	3	4	5	6	7
1	0	1	0	1	1	0	0
2	1	0	1	1	0	0	0
3	0	1	0	1	0	1	1
4	1	1	1	0	1	1	0
5	1	0	0	1	0	1	0
6	0	0	1	1	1	0	1
7	0	0	1	0	0	1	0

Figure 1: Adjacency Matrix Representation for the G2

2 Question 2

```
[akcaglavan17@ssh algo2]$ g++ -std=c++11 GA1_150170085_Q2.cpp -o GA1_150170085_Q2
[akcaglavan17@ssh algo2]$ ./GA1_150170085_Q2
Vertex 1--->Color 1
Vertex 2--->Color 2
Vertex 3--->Color 1
Vertex 4--->Color 3
Vertex 5--->Color 2
Vertex 6--->Color 4
Vertex 7--->Color 2

Number of different colors:4
Time in 0.028027 ms.
```

Figure 2: The Output for Question-2

3 Question 3

3.1 The Output for Question-3

```
[akcaglayan17@ssh algo2]$ g++ -std=c++11 GA2_150170085_Q3.cpp -o GA2_150170085_Q3
[akcaglayan17@ssh algo2]$ ./GA2_150170085_Q3
Vertex 4---> Color 1
Checking 3 ---> false
Checking 6 ---> false
Checking 1 ---> false
Checking 2 ---> false
Checking 5 ---> false
Checking 7 ---> true
Vertex 7 ---> Color 1
Vertices 4,7 are dropped!!

Vertex 3---> Color 2
Checking 6 ---> false
Checking 1 ---> true
Vertex 1 ---> Color 2
Checking 2 ---> false
Checking 5 ---> false  (since it is connected to 1)
Vertices 3,1 are dropped!!

Vertex 6---> Color 3
Checking 2 ---> true
Vertex 2 ---> Color 3
Checking 5 ---> false
Vertices 6,2 are dropped!!

Vertex 5---> Color 4
Vertices 5 are dropped!!

Well done!! All the vertices are colored.
Min color num:4
Time in 0.244699 ms.
```

Figure 3: The Output for Question-3

3.2 Time Complexity of GA2

As seen in Fig. 4, the sorting algorithm takes O(nlogn) time. Then, the outer while loop runs n times and the inner for loop also checks all the elements of the degreeList, that is, it runs n times. Because there are two nested loops, time complexity is O(n^2). Inner second for loop as it will cause lower complexity.

O(nlogn) + O(n(n+m)) = O(n^2) time complexity.

```
GreedyAlgorithm2()
{
    create degreeList to keep degree of each vertex
    sortByDegrees();                                  // O(nlogn)

    colorNumber = 1;

    while degree_list is not empty                    // O(n^2)
    {
        vertex v = first vertex of degreeList;
        vertex v colored with colorNumber c;
        add vertex v to dropList;

        for each element n in degreeList              // O(n)
        {
            if n is not a neighbor of and n does not have a neighbor of the same color with v
                Checking true;
                neighbor n colored with color c;
            else
                Checking failed;
        }
        for each element d of dropList                // O(m)
            remove i from degreeList;

        colorNumber++;
    }
}
```

Figure 4: Pseudo Code of GA2

4 Question 4

GA3 does not perform well for every possible scenario. If all neighbors of a picked vertex are colored, a new vertex cannot be randomly selected. In this case, uncolored vertices may remain in the graph.

A scenario where the algorithm is successful is as follows.

Vertex 6—Color 1

Vertex 7—Color 2

Vertex 3—Color 3

Vertex 4—Color 2

Vertex 5—Color 3

Vertex 1—Color 1

Vertex 2—Color 4

Number of different colors:4

A scenario where the algorithm fails is as follows:

Vertex 5—Color 1

Vertex 4—Color 2

Vertex 3—Color 1

Vertex 6—Color 3

Vertex 7—Color 2

Uncolored neighbor of Vertex 7 not found. Vertex 1 and Vertex 2 remained uncolored. Failed to color graph successfully.

5 Question 5

5.1 Adjacency Matrix Representation for the SNG

Figure 5: Adjacency Matrix Representation for the SNG

5.2 The Output for Question-5

```
Number of different colors:5
Time in 0.050745 ms.
```

Figure 6: The Output for Question-5 GA1

```
Well done!! All the vertices are colored.
Min color num:4
Time in 0.613411 ms.
```

Figure 7: The Output for Question-5 GA2

5.3 Compare GA1-GA2 in terms of execution time

Greedy algorithm 1 performed better than Greedy algorithm 2 in terms of execution time. As shown in Fig.6 and Fig.7, running time of GA1 is 0.050745 ms but GA2 is 0.613411 ms.

5.4 Compare GA1-GA2 in terms of Minimum Number of Colors

Greedy algorithm 2 performed better than Greedy algorithm 1 in terms of finding the minimum number of colors. As shown in Fig.6 and Fig.7, number of colors of GA2 is 4 but GA1 is 5.

6 Summary

Finally, for a given graph, GA1 offers a faster solution, but worse performs in terms of minimum color number. GA2 finds all vertices it can color with the current color. This finding takes longer time but performs better in terms of minimum color number. In this way, it always gives the GA2 minimum number of colors. GA3 is problematic because if all the neighbors of any vertex are colored, the remaining uncolored vertices in the graph remain uncolored. The answer to the question of which one is better depends on the aim of the program. If better performance in terms of time is desired, GA1 should be used. If a minimum number of color is desired, GA2 should be used.