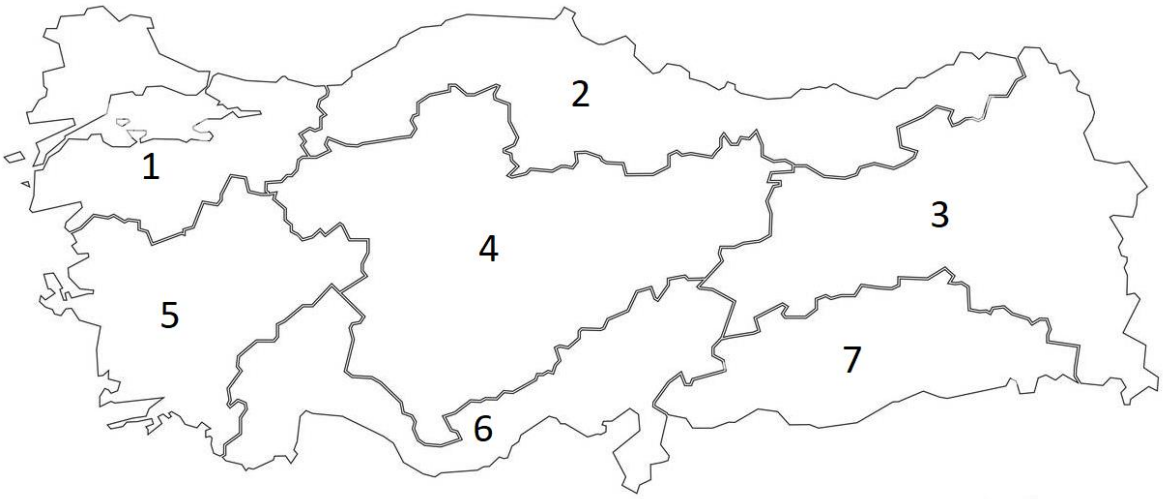
BLG336E - Analysis of Algorithms II

Project-2

Deadline: 22/04/2022 23:59

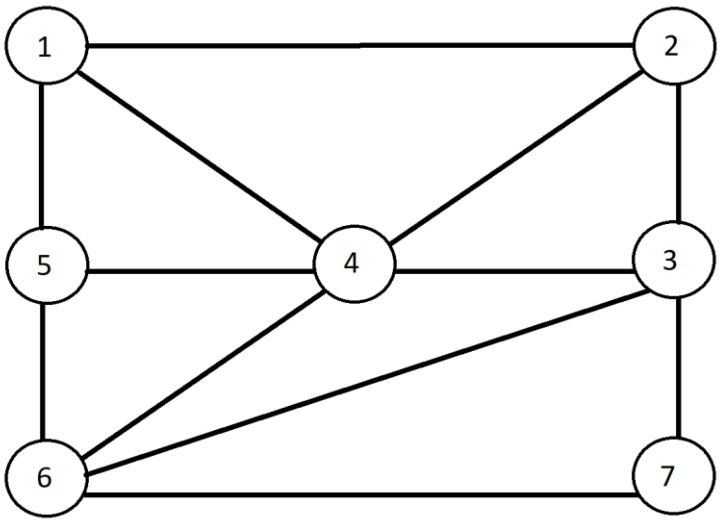
The Graph-1 below represents the Regions of Turkey (Regions are numbered 1 to 7) [1].

Graph-1 (G1):



In G1, each region is a vertex and it is assumed that adjacent vertices (regions) have a non-directional edge (see Graph-2).

Graph-2 (G2):



Please answer the following 4 questions according to the information above.

[1] https://www.alamy.com/stock-photo-turkey-outline-silhouette-map-illustration-with-regions-137108343.html

Question-1) Give the adjacency matrix representation for the G2 (**5p**).

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

Question-2) Write a program to implement GA1 given below (**15p**).

Greedy Algorithm-1 (GA1): The vertices will be visited in **ascending order** (1 to n)

(i.e. 1-2-3-4-5-6-7).

Step-1: Pick the **vertex 1** color it with **color 1**.

Step-2: Switch to the next vertex considering the ascending order.

Step-3: Repeat the following for the remaining vertices:

- For the currently picked adjacent vertex **color it** with the lowest numbered color such that there is **no** two connected vertices of same color.
- **Note that:**
  - **a.** The algorithm always moves one vertex to the next in ascending order, **even if the vertices are not connected.**
  - **b.** The lowest numbered color means that if you need to choose a color for a vertex and both Color 1 and Color 2 are available, always choose Color 1 which is the lowest.

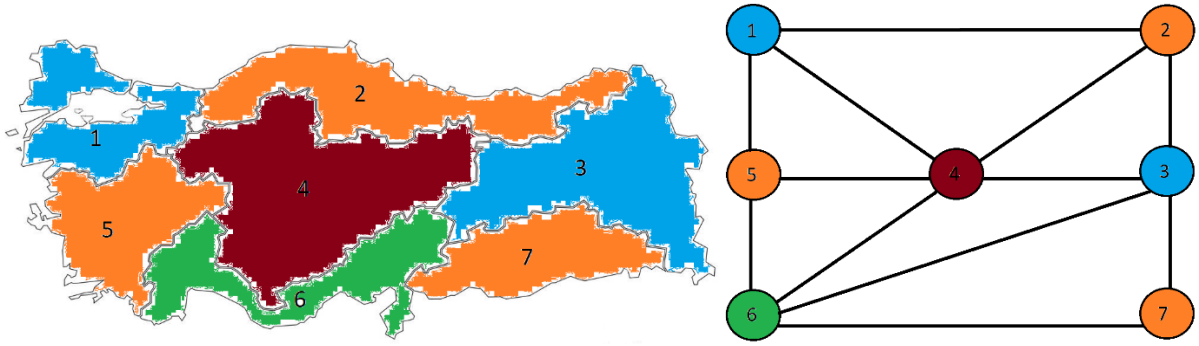
(i.e. you chose **vertex 1** initially and colored it with **color 1**. In this case, the **vertex 2** to be visited next cannot be colored with **color 1,** it will be colored with **color 2**. After that **vertex 3** is visited and colored with **color 1** if it is not connected to **vertex 1**.)

Scenario-1: 1-2-3-4-5-6-7

The **output** of the program:

```
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 ms.
```



Question-3)

a) Write the program to implement GA2 given below. (**15p**)
b) What is the complexity of the algorithm? Explain with pseudo code of your program (**5p**)

Greedy Algorithm-2 (GA2): The vertices will be visited according to the degree of each vertex. The vertex with the highest degree will be visited and colored first. (The traversal will be in descending order of degrees.)

GA2:

Step-1: Find the degree of each vertex.

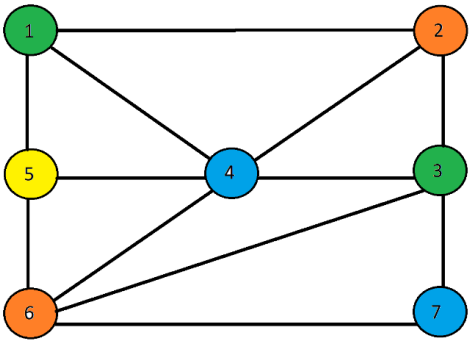
Step-2: List the vertices in **descending order of degrees** in a **degree_list.**

Step-3: Color the first vertex (having **the highest degree**) with **color 1**.

Step-4: In degree_list go down and color the vertices that are not connected to first vertex with the same **color 1**. Then drop the visited vertices from the **degree_list.**

Step-5: For the remaining vertices repeat the process with a new color until there are no uncolored vertices, keeping in mind the following rules:

- The vertices will always be in descending order of degree
- There will not be two adjacents with the same color



A possible order to visit: 4-3-6-1-2-5-7

The **output** of the program:

```
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 ms.
```

Question-4) Analyze the following greedy algorithm (GA3) and show whether the given GA3 performs well for every possible scenario? (**10p**)

- If not, please provide a counter scenario of GA3 failing.
- No code required. Just explain your answer.

vertex 7 -> color 1
adjacents of v7-> 3, 6
vertex 3 -> color 2
adjacents of v3 -> 2, 4, 6, =7
vertex 4 -> color 1
adjacents of v4 -> 1, 2, 5, 6 =7
v 1 -> color 2
adjacents of v1 -> 2, 5 =4
v2 -> color 2
adjacents of v2 -> =1, 3, 4
there is no "uncolored adjacent vertex out of of adjacent vertices"

Greedy Algorithm-3 (GA3):

1-) **Randomly** pick a vertex and color it.

2-) List all the adjacents of the picked vertex.

3-) Repeat the following for the remaining vertices:

- Randomly pick an uncolored adjacent vertex out of adjacent vertices and color it with the lowest numbered color such that any adjacent vertices will not be colored with the same color.
- Check for uncolored adjacent vertices of currently selected vertex.

- (i.e. if you choose **vertex 1** randomly, color it with **color 1**. Then the adjacent vertices **2, 4 and 5** can not be colored with **color 1**. Suppose you randomly select **vertex 5**, you will color it with **color 2** and then look for its uncolored adjacents. Note that two adjacent vertices with the same color is not allowed.)

Question-5): A social network graph (SNG) [2] has been given below that represents the relationship between people. Each people has been illustrated as nodes and numbered from 1 to 35.

Social Network Graph (SNG)

a) Give the adjacency matrix for **SNG**. (**5p**)
b) Code two new programs to implement **GA1** and **GA2** on **SNG**. (You can modify the programs in Question-2 and Question-3) (**2*10p=20p)**
c) Which greedy algorithm performed better in terms of execution time? Output the execution times in your program. (**5p**)
d) Which greedy algorithm performed better in terms of finding the minimum number of colors? Show it in your program output. (**5p**)

- The report will be graded out of **15p**. Try to prepare well-designed reports. Check Report keynotes at **page-7** for more details.

[2] https://www.dreamstime.com/royalty-free-stock-photo-people-social-network-image28316315

SOME KEYNOTES

Coding Part:

- Plagiarism will be checked and those detected to be cheating will receive **zero** and ITU disciplinary rules will be applied.
    - Online resources will be checked.
    - Student code files will be checked.

  Implement the programs on your own.
- You should write your code in C++ language and try to follow an object-oriented methodology with well-chosen variables, methods, and class names.
- Your code must be executable in Linux operating systems. Therefore, before uploading to Ninova, make sure that your codes are able to run without any errors. You can test your code through ITU's Linux Server (you can access it through SSH).
- Your code should **contain comments with the explanation** of the relevant line/part when necessary. Code without any comments will receive lower grade.
- You are free to use external C++ libraries such as Standard Template Library (STL).
- Name your code files in the following format:
    - GA1_YourID_Q2.cpp
    - GA2_YourID_Q3.cpp
    - GA1_YourID_Q5.cpp
    - GA2_YourID_Q5.cpp

Report:

- Your report will be like an answer sheet but it should **include:**
    - a cover page (Name-Surname-ID-Department, etc.)
    - table of the contents
    - Answer of the Question-1 (adjacency matrix).
    - Screen shot (SS) of the output for Question-2.
    - SS of the output for Question-3 **a)**.
    - For Question-3 **b)** write the pseudo code of your program and calculate the time complexity.
    - For Question-4, give examples of different scenarios and explain your answer.
    - For Question-5 **a)** provide adjacency matrix.
    - Answer Question-5 **c)** and **d),** provide SS of the outputs**.**
    - Finally, **discuss** whether GA1/GA2 always give the minimum number of colors for any given graph. Compare the performances of GA1 and GA2? What is the problem with GA3? Which one is better? Why?

- Name your report file in the following format:
    - Project2_YourID_YourName.pdf

*If you have any questions, please contact T.A. Uğur AYVAZ via ayvaz18@itu.edu.tr