MATH 1302-01 Discrete Mathematics

Instructor: Professor Yazali Kumar

Name: Ryohei Hayashi

Written Math Assignment Unit 7

**1. In a hostel, there are around ‘n’ number of students (assume a number above 100 for n) with rooms categorized as triple, double, and single occupancy. Explain how you will represent the above data as a graph.**

**You need not create or share the graph but make sure to include justification for the below questions:**

**(i) Will the graph be a simple graph or a multigraph?**

**(ii) Will it have loops?**

**(iii) What is the possible maximum and minimum degree for each student?**

**(iv) If all problems are represented in the form of graphs, won’t the problems be easily visualized and solved? (v)Can we represent every problem with a graph? Explain the reason by considering an example of a situation that has 11 vertices such that the degree of each vertex is 11.**

I have come up with a way to represent as a graph the data of "n" students who stayed at the hostel and who were categorized into three-person rooms, two-person rooms, and one-person rooms. To represent this data as a graph, I take the following approach

(i) Representation of vertices

Each vertex of the graph represents a student. That is, there are n vertices.

(ii) Representation of edges

An edge represents the room-sharing relationship between students. For example, if a student stays in a three-person room, there is an edge between each of the three students. If a student is staying in a single room, he/she will not have an edge.

(iii) Representation of room type

Room types are represented by the properties of the edges and the degree of the vertices. For example, the vertex of a student staying in a single room has degree 0 because it is not related to other students. Similarly, the degree of a student staying in a double room is 1, and the degree of a student staying in a triple room is 2.

(iv) Graph Characteristics

Simple or multiple: The graph is represented as a simple graph. This graph is represented as a simple graph since each student is assigned to a specific room type and there can be no more than one relationship. A multigraph is used when there are multiple edges between two vertices, which is not the case in this scenario.

Are there loops: There are no loops in this graph. Loops occur when a vertex is associated with itself, but in this scenario, there are no students associated with themselves.

Maximum and minimum degree possible for each student: The maximum degree is 2 for a 3-person room. Similarly, the minimum degree is 0 for a single room.

The goal of this question was to use a graph to categorize the students staying at the hotel and the type of room. While graphs are good for visualizing relationships and connections, it is difficult to solve all problems using graphs because there are cases where patterns such as this problem are difficult to represent or solve. I believe that it is desirable to use the appropriate method to solve each problem.

For example, consider the case of requirement (v) of the question. The following relationship holds for the number of vertices n and the maximum degree of its graph

Maximum degree = n-1

For a graph with 11 vertices, it is not feasible to create a graph-like requirement (v) because the maximum degree is 10.

**2. A University is conducting a conference for two days on different subjects for students pursuing their higher education. Your task is to create a scheduling slot scheduling model for the conference sessions (based on the subjects attended by students) using graph coloring. Draw the graph for the same and answer the chromatic number of this problem of scheduling time slots with the number of subjects your choice.**

To create a schedule for a conference session using graph coloring, this can be implemented by representing the perimeters in the graph as vertices and the conflicts between subjects as edges.

In this case, I will proceed with the creation of vs. Usukeju=r assuming that the following conflicts occur for the five subjects of Mathematics, Physics, Computer Science, Chemistry, and Biology.

* Mathematics - Physics
* Math - Computer Science
* Math - Chemistry
* Physics - Chemistry
* Chemistry - Biology

To schedule a conference session, timeslots must be assigned so that subject conflicts do not occur in the same time slot. This is the same task that results in coloring the vertices of a graph so that adjacent vertices do not have the same color. Various algorithms exist for determining the number of colors in a graph, but in this case, I proceeded by determining the colors from the vertices with the highest degree, so that adjacent vertices are not assigned the same color. As a result, it is possible to color the graph with three colors as shown below.

* Mathematics: Color A
* Physics: Color B
* Computer Science: color B (can be the same color as Physics because it does not conflict with Physics)
* Chemistry: color C (must be other than color A or B because it conflicts with math and physics)
* Biology: color A (must be other than color C because it conflicts with only Chemistry)

Thus, in the example I have created, the conference can be held if 3 time slots exist over 2 days. However, since this is the minimum number of timeslots, it is possible to increase the number of timeslots while avoiding conflicts.

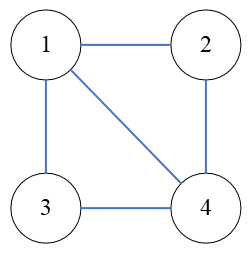
**3. Explain Euler and Hamiltonian cycles, and provide one simple counter example for each. Find the Euler circuit/path and Hamiltonian cycle/path for the given graph G.**

Let's start by describing each of these circuits.

Eulerian Circuit: An Eulerian circuit is a closed path that passes through all edges only once and returns to its original vertex. For an Eulerian circuit to exist, the graph must be connected and the degree of all vertices must be even.

Counter-example:

Consider the following graph with 4 vertices.

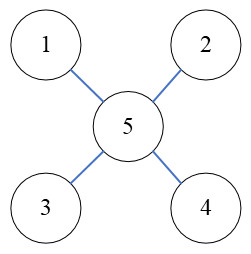


This graph has degree 2 for vertices 2 and 3 and degree 3 for vertices 1 and 4. This graph is not an Eulerian circuit because it violates the condition that the degree of all vertices is even.

Hamiltonian Circuit: A Hamiltonian circuit is a closed path that passes through every vertex of a graph only once and returns to the original vertex. It differs from an Eulerian circuit in whether it focuses on edges or vertices.

Counter-example:

Consider the following graph with 5 vertices.



For this graph to be a Hamiltonian circuit, it must pass through all vertices 1-4, but it cannot pass through vertex 5, given the condition that it passes through it only once. Therefore, this graph is not a Hamiltonian circuit.

Next, consider the given graph G.

Whether or not an Euler circuit exists:

To check if it is an Euler circuit, I first check the degree of all vertices.

Degree sequence = (2,4,3,4,4,3,2)

I know that graph G does not have an Euler circuit because it violates the condition that the degree of all vertices is even.

Whether or not a Hamiltonian circuit exists:

The only way to be sure that a Hamiltonian circuit exists is to check the path. I checked for graph G and found that a Hamiltonian circuit exists. The path is as follows.

Vertex 1 -> Vertex 2 -> Vertex 3 -> Vertex 5 -> Vertex 4 -> Vertex 6 -> Vertex 7

**4. Explain the spanning tree. Find at least two possible spanning trees for the following graph H and explain how you determined that they are spanning trees. Draw a bipartite graph from any one of the two spanning trees that you found.**

A spanning tree represents a subgraph that contains all vertices of a graph and has no cycles. The following are the characteristics of spanning trees.

Contains all vertices: A spanning tree contains all vertices of the original graph.

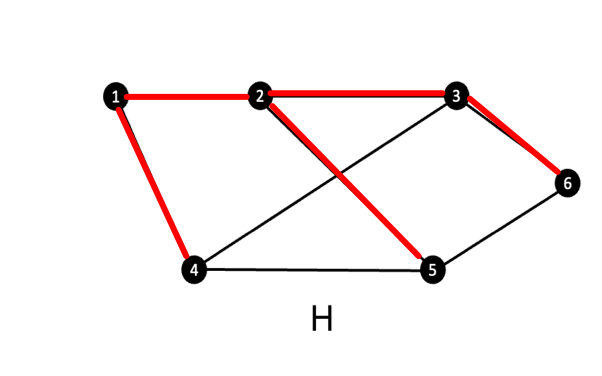
Minimal edges: A spanning tree contains the minimum number of edges needed to connect a graph. That is if the number of vertices is n, the number of edges is n-1.

No cycles: Spanning trees have no cycles. If there were cycles, it would not be a tree.

Connected: A spanning tree is connected, so there is always a path between any two vertices.

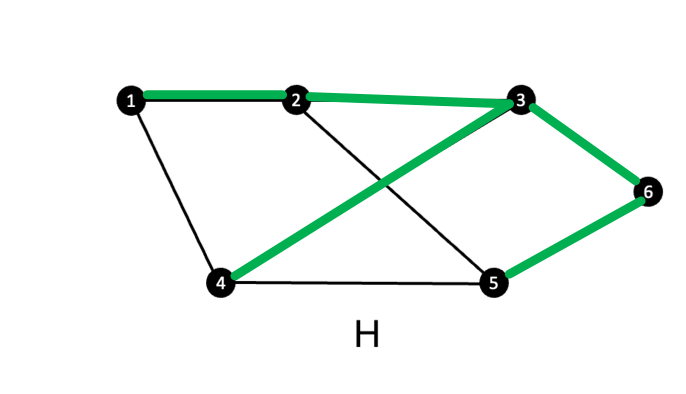
Next, I consider a spanning tree for a given graph H. Two spanning trees are considered.

The first (red)



This spanning tree was created to have a tree structure with vertex 1 as the root. I judged it to be a spanning tree based on the following three points: all vertices are included, the relationship between the number of vertices and the number of edges, and the absence of cycles.

Second (green)



This spanning tree was created to make it a tree structure with vertex 6 as the root. Similar to the first example, I can say that it is a spanning tree because it satisfies the three conditions.

Next, I consider a bipartite graph for the first example. A bipartite graph is a special kind of graph that can be partitioned into two independent sets of vertices. To create a bipartite graph for a spanning tree, one can assign any set to a root and then assign different sets to its leaves. This can be done by implementing this until the last leaf. In this case, I used the policy of splitting the graph into sets A and B.

