Group HW #04

Abby Omer, Adi Sudhakar, Antoinette Tan, Dasha Chadiuk

November 1, 2021

- ALL DID ALL
- Resources used: We referenced the textbook and the in-class notes.
- Group goal this week: To collaborate in a way that is efficient.

Team Contract

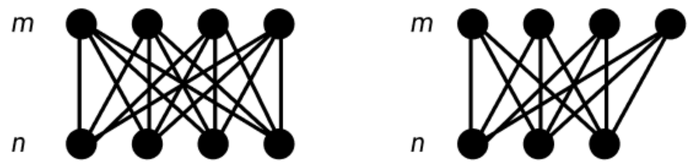
All members (Abby, Adi, Antoinette, and Dasha) will meet on Saturday 9am each week to work on group problems. To prepare for these meetings, everyone is expected to attempt each homework problem before the meeting. In addition, one person each week will write up the report to be turned in. The Latex document will be sent to everyone in the group the Sunday morning before the homework is due so that each group member can look over the report before it is submitted. This task of writing up the report will be assigned to people on a rotation basis for each group homework.

Problem 1

8.2.36: A simple graph is called regular if every vertex of this graph has the same degree. A regular graph is called n-regular if every vertex in this graph has the degree n. For which values of m and n is $K_{m,n}$?

Solution: Recall the definition of the complete bipartite graph, $K_{m,n}$: it is the graph that has its vertex set partitioned into the two subset of m and n vertices. Edges are drawn between vertices if and only if the first vertex is in the subset of m and the second vertex is in the subset of n. Now, recall the definition of a regular graph: all vertices have the same degree. This is only possible if the number of vertices in the first subset is equal to the number of vertices in the second subset. **Therefore, for $K_{m,n}$ to be normal, m must be equal to n.**

Figure 1: Left: $m = n$, all vertices are same degree. Right: $m \neq n$, vertices have different degrees.



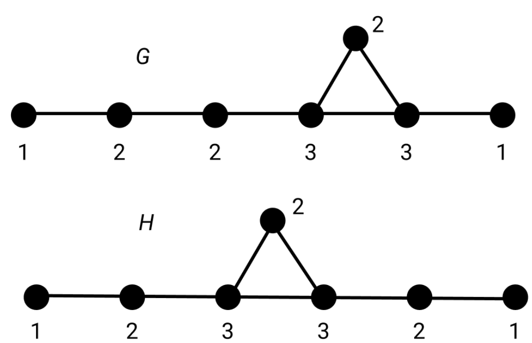
Problem 2

8.3.69: A devil's pair for a purported isomorphism test is a pair of non-isomorphic graphs that the test fails to show are not isomorphic. Find a devil's pair for the test that checks the sequence of degrees of the vertices in the two graphs to make sure they agree.

Solution: Consider these two simple graphs below. Each graph has 5 vertices and 6 edges. Things we must check to prove a devil's pair:

1. **Degree Sequences are Identical** - is the monotonic, non-increasing sequence between both graphs' valencies the same?
2. **Isomorphism is Violated** - are mappings between vertices inconsistent between both graphs?

Figure 2: Example of a Devil's Pair for the above test. Degree of each vertex is labeled



In both graphs the degree sequence is {3,3,2,2,2,1,1}, and there are 7 edges and 7 vertices. These graphs would pass the test that checks the sequence of degrees, however we can see the two graphs are not isomorphic. In the lower graph, H, there are 2 vertices with a degree of 3 that both share an edge with 2 vertices with a degree of 2. This is not possible for the top graph, G, where one of the vertices with degree 3 can only ever share an edge with a single vertex with a degree of 2.

Graph Theory Investigations

Euler & Hamilton Circuits & Paths: An Euler path is a simple path containing every edge of the graph once and start and end vertices. An Euler circuit is a simple circuit that contains every edge of the graph once and ends on the starting vertex. Euler found that a connected multigraph has a Euler circuit only if each vertex has an even degree. Each time the path passes through a vertex, it must enter through one and exit through another. Euler also found that a connected multigraph had a Euler path but not a circuit only if it has exactly two vertices of odd degree, allowing for the path to start at one vertex and end at another. A Hamilton path is a simple path that contains every vertex of a graph and ends on a different vertex from the start, while a Hamilton circuit is a simple circuit that contains every vertex of a graph once and ends on the start vertex. Driac found that a Hamilton circuit exists if G has n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$. Ore found that a Hamilton circuit exists if G has n vertices with $n \geq 3$ such that $deg(u) + deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G. Overall, the more edges a graph has, the more likely a Hamilton circuit exists. Hamilton circuits are useful in finding the shortest path a "travelling salesman" should take by finding the circuit with smallest total weight of its edges.

Shortest Path Algorithms: An interesting way to solve shortest path problems is to model them using graphs with edges that have assigned weights. These weights can represent distances, time, money, etc. depending on the type of problem you are trying to solve. As mentioned above in the Hamilton circuit section, we can find the shortest path that a salesman should take between cities if we model the cities as vertices and the paths he could take as the weighted edges. We can find the Hamilton circuit that minimizes the distance (total sum of the edges he travels on) between cities. One algorithm that can be used to find the shortest path between vertices in a weighted graph is Dijkstra's algorithm. It works by finding the shortest paths from a starting vertex to all the other vertices. While it is very tedious to do by hand, a computer can easily find all the shortest paths using the algorithm. One application of this algorithm is in digital mapping services such as Google Maps. When you put in your starting location and your destination in Google Maps, it uses the shortest path algorithm to calculate the minimum distance between the two locations. The algorithm has certain drawbacks - it does a blind search wasting time and resources; it also cannot handle negative edges.

Planar Graphs: A graph is called planar if it can be drawn without any edges crossing. Some planar graphs may be drawn with edges crossing (though they can be represented otherwise). Planarity of a graph has useful applications like: organization of public transportation routes, electrical transmission lines, utilities and PCB design. The problems listed above are not unidimensional, and therefore a planar graph by itself cannot account for factors like energy consumption, traffic, etc.That said, planar graphs can provide a good 0th order approximation for solving and optimizing these problems. Planar graphs have 3 characteristics - edges, vertices, and regions. Euler showed that the number of regions (including unbounded) in a planar graph is constant, as are the number of edges and vertices - this relationship for a graph G can be expressed as $r = e - v + 2$. A non-planar graph by definition needs at least 1 edge to cross another. Kuratwoski claimed that since the graphs $K_{3,3}$ and $K5$ are non-planar, any non-planar graph must have a subgraph that is homeomorphic to either $K_{3,3}$ or K_5.

Graph Coloring: One interesting fact about graph coloring is that the chromatic number of a planar graph is no greater than four. This is the Four Color Theorem and was proven by

K.Appel and W. Haken. One application of graph coloring is scheduling; a graph can be created with the vertices representing an event, and an edge between two vertices representing that there is a common person in the events. A scheduling of the events corresponds to a coloring of the associated graph. Each time slot for an event is represented by a different color. To find the number of time slots needed, the chromatic number of the graph can be used. Another application of graph coloring is index registers. In efficient compilers, the execution of loops is sped up when frequently used variables are stored temporarily in index registers in the CPU, instead of the regular memory. A graph can be created with the vertices representing a variable in the loop, and the edges between two vertices are in the index registers at the same time during the execution of the loop. The number of index registers needed can be found with the chromatic number of this graph.

Trees: One application of trees are binary search trees. In the tree, each child of a vertex is designated as a right or left child, no vertex has more than one of either, and each vertex is labeled with a key, which is one of the items. Vertices are assigned keys so the key of a vertex is both larger than the keys of all vertices in its left subtree and smaller than the keys of all vertices in its right subtree. A practical example: we can locate an item if it is present and add it to the tree if it is not. If a binary search tree is balanced, locating or adding an item requires no more than $\lceil (log(n+1)) \rceil$ comparisons. Another application of trees are prefix codes; an application would be using bit strings to encode the letters of the English alphabet. Prefix codes can find a coding scheme of the letters so that, when data are coded, fewer bits are used while still making sure that no bit string corresponds to more than one sequence of letters. This application makes it so that the bit string for a letter never occurs as the first part of the bit string for another letter. We can construct a prefix code from any binary tree where the left edge at each internal vertex is labeled by 0 and the right edge by a 2 and where the leaves are labeled by characters. Characters are encoded with the bit string constructed using the labels of the edges in the unique path from the root to the leaves.

For Group HW 5

Our team decided to further investigate Shortest Path Algorithms for the CDD. We chose this because all of us investigated it for part A and are interested in its practical uses and non-obvious applications.

Team Report

Our group goal this week was to work efficiently. Like previous group homeworks, we attempted each problem on our own and met Saturday morning to discuss them. We took turns explaining the problems to each other, going through the methods we used to get to an answer. If any group members used different methods (or had different answers) we talked through all methods until we agreed on the solution. After the meeting, one person typed up the LaTeX report and then others reviewed it and made changes. This strategy allowed us to efficiently complete the group homework and to learn from each other. We feel that we have met our team goal for this week.