Ethan Wong
11/30/20

Math 61 HW Week 7

1. There aren't many values for m and n where a complete bipartite graph $K_{m,n}$ can be planar.
$K_{1,n}$ will be planar for all n as it forms a star graph.
Ex:

Regardless of n, the graph will still be planar.
$K_{2,n}$ is also planar by similar logic as above. It would basically be two connected star graphs.
All other scenarios of graphs will be nonplanar because they will deal with $K_{3,3}$, which is nonplanar. Since all other graphs will contain $K_{3,3}$ as a subgraph, they will all be nonplanar by Kuratowski's theorem.

3. Spanning trees are frequently used in the networking aspect of computing. They are also used for electrical applications such as landlines. For example, say that a corporation has many locations across the United States. The CEO then wants to set up a private phone line that connects all the locations together. The phone provider charges varying amounts to connect different cities together (this would correspond to the weights in the graph). Obviously the smartest idea would be to connect all the offices with minimal cost. This is a classic application of a spanning tree in a real-world situation.

4. breadth_first_search (V, E):

```
S = (v_1)
V' = {v_1}
E' = ∅
while (true):
  for each x ∈ S, in order,
    for each y ∈ V - V', in order,
      if (x,y) is an edge
        add edge (x, y) to E' and y to V'
  if no edges were added
    return T
  S = children of S ordered consistently with original vertex ordering
```

We can prove the breadth-first-search (BFS) algorithm is correct using a spanning tree for the graph G.

A spanning tree of the simple graph G is a subgraph of G that ① contains all vertices of G and ② is a tree.

In the algorithm, edges and vertices of G are added to T in every step. No other edges and vertices get added. This implies that T is going to be a subgraph of G.

Only edges (x,y) with $x \in S$ and $y \in V - V'$ are added in the algorithm. This implies that an edge is only added to T when one of the vertices is in the graph already and the other vertex isn't in T yet. Because of this fact, we then know that a cycle can never be formed within T. Because T is connected (G is connected, which means T must be connected) and there are no cycles within T, we know that T is a tree.

The algorithm only stops when $V = V'$ — this means that T will contain all vertices of G at the end of the algorithm. Therefore, T must be the spanning tree of G and the breadth-first-search algorithm has been proven to be correct.

2. Show that if a graph contains a cycle it contains a cycle with no repeated vertices.

By definition, a cycle is a non-empty (not length 0) trail from v to v with no repeated edges.

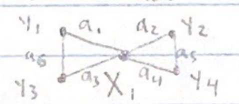
A cycle without repeated vertices other than the requisite first and last vertex is known as a simple cycle.

Let's assume that there's a cycle with a repeated vertex X_1 on graph G.

The degree of X_1 is 4 because the cycle passes through it twice. This means there are also 4 edges incident to X_1. 2 edges lead into X_1 (call these a_1 and a_2) and 2 edges go away from X_1 (call these a_3 and a_4).

Our cycle would look like this:

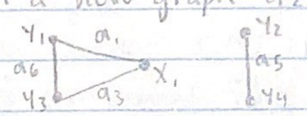
$\underline{G}$:



cycle : $(X_1, a_1, X_1, a_4, y_4, a_5, y_2, a_2, X_1, a_3, y_3, a_6, y_1)$

We can form a new graph G_2 by getting rid of a_2 and a_4.

$\underline{G_2}$:



Now we observe that a simple cycle emerges on the left side of G_2 if we do this.

This proves that any given graph with a cycle will contain a cycle with no repeated vertices, or a simple cycle. This can be shown by removing the incident edges in the graph that cause vertices to repeat.

Basically, if a given vertex X_1 occurs twice in a cycle, we can delete/remove the part of the cycle that goes from X_1 back to X_1. This process can be repeated until there are no more repeated vertices and there is a simple cycle now in the graph.

For example, if there was a repeated vertex, the cycle leaves that vertex on a distinct edge. It will then come back to that same vertex via another edge. This smaller section would also be considered a cycle since the overall graph is connected. Using what I said previously, removing the two edges from the above example would eliminate the repeated vertex, yielding a simple cycle with no repeated vertices that can be shown in the larger cycle.