Module 2 Advanced: Graph Proofs

1. *Kruskal's algorithm* begins by adding the smallest edge in the graph to the solution (and never looking back). Let $e = (u, v)$ be a minimum-weight edge in a connected graph G. Show that $e = (u, v)$ belongs to some minimum spanning tree of G. *HINT: Use a proof by contradiction. Note that e will eventually connect two smaller spanning trees together. If e is NOT in the solution, than something else IS. Show that this leads to some kind of contradiction.*

2. This problem is about robots that need to reach a particular destination. Suppose that you have an area represented by a graph $G = (V, E)$ and two robots with starting nodes $s_1, s_2 \in V$. Each robot also has a destination node $d_1, d_2 \in V$. Your task is to design a schedule of movements along edges in G that move both robots to their respective destination nodes. You have the following constraints:

   1. You must design a schedule for the robots. A schedule is a list of steps, where each step is an instruction for a single robot to move along a single edge.
   2. If the two robots ever get close, then they will interfere with one another (perhaps start an epic robot fight?). Thus, you must design a schedule so that the robots, at no point in time, exist on the same or adjacent nodes.
   3. You can assume that s_1 and s_2 are not the same or adjacent, and that the same is true for d_1 and d_2.

   Design an algorithm that produces an optimal schedule for the two robots. What is the runtime of your algorithm? How would the runtime change as the number of robots grows?

3. This question is about the *depth-first search tree* and *breadth-first search tree* generated from a given **connected** graph G. Recall that these trees are formed by including the subset edges from E that are traversed to first discover each node in the respective search. With this in mind, prove the following claim:

   If T_d is the depth-first search tree generated by running DFS on G rooted at some node u, and T_b is the breadth-first search tree generated by running BFS on G rooted at that same node u, then $T_d = T_b \rightarrow G = T_d = T_b$. In other words, if BFS and DFS produce the same tree, then the entire graph G was already a tree.

   Hint: Try proving this by induction on G. When G is just one node, the claim is pretty trivial. What if G is two nodes? Three nodes? etc.