

Graphs - Advanced: Graph Proofs

1. **Warm Up:** Let G be an undirected graph with n nodes (let's assume n is even). Prove or provide a counterexample for the following claim: If every node of G has a degree of at least $\frac{n}{2}$, then G must be connected.
2. Prove that the edge e that Prim's algorithm selects at each point in the algorithm must be in the minimum spanning tree. Specifically, assume the e is not in the minimum spanning tree and use an exchange argument to show that it must be.
3. Formally prove that the *Bi-Partite Matching* algorithm we saw in class is optimal (i.e, it always finds the optimal matching between nodes in the bi-partite graph). *HINT: Assume the algorithm is not optimal and show that you must be able to still find an augmenting path through the network, contradicting your assumption that max-flow terminated.*
4. Suppose you are given a list of potential flights $F = \{f_1, f_2, \dots, f_n\}$ that your airline wishes to serve. You can think of these as flight objects. Each flight has several fields including starting airport, ending airport, departure time, and arrival time. Develop an algorithm that, given flights F and a value k , determines whether it is possible to serve all of the flights in F with at most k planes. You must adhere to the following additional constraints:
 1. Plane maintenance takes m minutes. No plane can depart less than m minutes after arriving at an airport.
 2. A plane can fly between any two airports. For example, if you have two flights you want to serve in F (LA to BOS, NYC to DC) then we might choose to fly a plane from LA to BOS, then fly from BOS to NYC (even though that leg isn't in F) so that this plane can then serve NYC to DC.
 3. You are given another parameter l . If you are splicing in an extra flight (see item above) then you must have at least m time for maintenance and l time for the other flight (yes, this is a pretty simple and unrealistic assumption, but is fine for our purposes).
 4. A plane does not need to end its day at the same airport from which it started. You can assume that all flights are during the day (between 6am and 12 midnight), and that there is an overnight flight at the end of the schedule back to the first departing airport before the next day begins.
5. 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?