Homework 5: Sorting, Graphs, and Network Flow

1. [2 points] Prove the max-flow min-cut theorem. In particular, show that if you have some flow f for a flow network G, then the following three statements are equivalent:

    1. There exists a cut $C = (A, B)$ such that $capacity(C) = value(f)$.
    2. f is the maximum flow.
    3. There does not exist an augmenting path in the residual graph G' calculable from G using flow f.

2. [4 points] **Fixing the Dam:** OH NO! The dam is leaking and might break, drowning a nearby vet clinic filled with sick puppies! Let's use the power of algorithms to save those puppies!

    The wall of the dam is rectangular and has specific points with holes. You've been told you can model this as a grid of size n by m (let's call it $wall$). Every position $wall[i][j]$ is either ok (meaning there is no hole here) or broken (meaning there is a hole here).

    You have special repairing bricks that you can use to patch the wall. The problem is that each brick is exactly $2x1$ in size. Thus, any brick must fill in two adjacent holes in the wall (a brick cannot fill in only one hole and two bricks cannot overlap). Design an algorithm that given the state of the wall, calculates where to place bricks so that all holes are patched, or returns false if it is not possible to fill in the holes in the dam perfectly.

3. [4 points] 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?