For each of the problems below, transform the input into a graph and apply a standard graph algorithm that you've seen in class. Whenever you use a standard graph algorithm, you ***must*** provide the following information. (I recommend actually using a bulleted list.)

- What are the vertices? What does each vertex represent?
- What are the edges? Are they directed or undirected?
- If the vertices and/or edges have associated values, what are they?
- What problem do you need to solve on this graph?
- What standard algorithm are you using to solve that problem?
- What is the running time of your entire algorithm, *including* the time to build the graph, *as a function of the original input parameters*?

---

1. You are planning the seating arrangement for a wedding given a list of guests, V . For each guest g you have a list of all other guests who are on bad terms with them. Feelings are reciprocal: if h is on bad terms with g, then g is on bad terms with h. Your goal is to arrange the seating such that no pair of guests sitting at the same table are on bad terms with each other. There will be only two tables at the wedding. Give an efficient algorithm to find an acceptable seating arrangement if one exists.

2. Plum blossom poles are a Kung Fu training technique, consisting of n large posts partially sunk into the ground, with each pole pi at position (xi, yi). Students practice martial arts techniques by stepping from the top of one pole to the top of another pole. In order to keep balance, each step must be more than d meters but less than 2d meters. Give an efficient algorithm to find a safe path from pole ps to pt if it exists.

3. The knight's tour is a classic problem that asks given a $n \times n$ board, and a knight with a particular starting position (x, y), is there a sequence of moves that the knight can make so that it visits every square exactly once. Divise an algorithm that finds a knight's tour (if there is one). Don't worry about your algorithm being efficient.