In today's lecture, we're going to discuss using graphs to solve problems. This is known as a "reduction" but don't let the fancy name fool you – you are familiar with the basics of the technique by knowing about software libraries.

The first problem used to be a homework problem in this class.

1. We have two containers: one has a capacity of three gallons of water, the other five gallons. Both are initially empty, although we have access to a large water fountain. We can take a non-full container and fill it completely up with the water, or we can completely empty a container (with water in it) into the fountain, or we can pour the contents of one container into the other until either the first is empty or the second is full. Our goal is to find a sequence of actions we can take to end up with exactly four gallons of water in one container (and none in the other container).

   (a) How can we represent this as a graph? You should specify what the vertices and edges represent, how many vertices there would be, and whether the edges are directed or undirected.

   (b) If you drew out the entire graph, how would you find the solution to the problem? Explain specifically what you would be looking for; you do not need to provide code or an algorithm, just a statement of what piece(s) of the graph your code would try to find. For example, if you would look to see if there are at least ten vertices, state that (this is not the correct solution).

2. A number maze is an $n \times n$ grid of positive integers. A token starts in the upper left corner; your goal is to move the token to the lower-right corner. On each turn, you are allowed to move the token up, down, left, or right; the distance you may move the token is determined by the number on its current square. For example, if the token is on a square labeled 3, then you may move the token three steps up, three steps down, three steps left, or three steps right. However, you are never allowed to move the token off the edge of the board. Your goal is to find the minimum number of moves required to solve a given number maze, or to correctly report that the maze has no solution.

   For example, in the following maze, we can solve this with eight moves: right, down, left, right, up, left, right, down.

   Describe how to use a graph to solve this problem. A complete answer includes a description of how to form a graph from an arbitrary number maze as well as how to find a solution to the number maze using the graph, or to report that none is possible.

3	5	7	4	6
5	3	1	5	3
2	8	3	1	4
4	5	7	2	3
3	1	3	2	

The following problem is a good challenge and reinforcement activity, but we will not have time to cover it in lecture. I am presenting the question as if it were a homework or exam question, although you **do not** need to turn this in for credit. This would be a great question to attempt, first on your own, and then to discuss with your study group.

Suppose you are given an unweighted directed graph $G = (V, E)$ which may or may not have cycles. The vertices, however, are all designated as azure, navy, or gold. This may or may not be a valid 3-coloring of the graph (and, for this problem, it does not matter if it is). The **secluded factor** of any given vertex v is the **largest** of the following three quantities:

- The shortest distance from any gold vertex to v
  (that is, among all the gold vertices, whichever has the shortest path from it to v is the quantity for this item)
- The shortest distance from any azure vertex to v
- The shortest distance from any navy vertex to v

If we cannot get to any particular vertex from at least one gold vertex, at least one navy vertex, and at least one azure vertex, then that vertex is *infinitely* secluded.

Given such a graph, describe how you would find a vertex with the *smallest* secluded factor. Give and briefly justify the running time of your approach. For full credit, your approach should have linear running time. You may make any reasonable assumption about how the graph is represented that you would like; if it affects your algorithm, write the assumption in the answer space.