1. Imagine that you have just been confirmed as the head of the super-secret spy agency SRU (Spies R Us). There are a number of countries that you wish to spy upon. These countries are of various sizes and speak a variety of languages. SRU has a number of spies each of whom speaks some languages, but not others. For each country you know how many spies are required to cover its entire territory and the language spoken. For each spy you know what languages he speaks. Obviously, you do not want to assign a spy to a country where he doesn’t speak the language. Give a good algorithm to assign spies to countries and find the time complexity of your algorithm.

This is a matching problem. For each spy create a node. If a country $c$ requires $n_c$ spies then create $n_c$ nodes for that country. Connect a node representing a spy to a node representing a country if the spy speaks the appropriate languages. Converting this problem to a network flow problem (by adding a source, a sink, edges from the source to each spy node, edges from the sink to each country node, and make all edge capacities 1). and running the augmenting path algorithm described in class yields a polynomial time algorithm $O(n^3)$ because each augmentation takes time at most $O(n^2)$ and there are at most $n$ augmentations.

2. Imagine that you have a large number of sweaters in a variety of styles, colors, and sizes. There are a large number of people each of whom like some of the sweaters (and are not interested in the others - different people like different sweaters). You want to distribute as many sweaters to these people as possible with the restrictions that no one is given more than 3 sweaters and no one is given a sweater that they don’t like. Give a good algorithm to solve this problem and find the time complexity of your algorithm (the more efficient your algorithm the better).

Create a 3 nodes for every person. Create a node for every sweater. Connect a “person node” to a “sweater node” if that person likes that sweater. Solve the bipartite matching problem on this graph which we know can be solved in time $O(n^3)$. This gives an assignment of sweaters to people that maximizes the number of sweaters assigned.

3. Let $G$ be an connected, undirected graph with $n > 0$ vertices and at least $n$ edges. Show that $G$ must have a cycle.

(Induction) A graph with $n = 1(2)$ nodes cannot have 1 (2) edges. A graph with 3 nodes and 3 edges must be a triangle and therefore has a cycle.

Assume that every graph with $m$ or fewer vertices and at least $m$ edges has a cycle. Let $G$ be a graph with $m + 1$ vertices and at least $m + 1$ edges. If $G$ contains a vertex with degree 1, removing this vertex and edge yields a graph with $m$ nodes and at least $m$ edges. By the inductive hypothesis this graph has a cycle and therefore $G$ contains the same cycle. If every node of $G$ has degree has at least 2 then start to perform a depth first search on $G$. At some point a vertex is reached which is adjacent to a vertex $v$ already visited (the initial forward search cannot terminate because it reaches a node with degree 1, since there are no nodes with degree 1). The portion of this path between the first visit to $v$ and the second visit to $v$ is a cycle.
4. What is the maximum number of edges a directed graph with n vertices can have and still not contain a directed cycle? Explain.

The number is $n(n - 1)/2$. Any directed graph has at most $n^2$ edges. However, since the graph has no cycles it cannot contain a self loop and for any pair $x, y$ of vertices at most one edge from $(x, y)$ and $(y, x)$ can be included. Therefore the number of edges can be at most $(n^2 - n)/2$ as desired.

It is possible to achieve $n(n - 1)/2$ edges. Label $n$ nodes 1, 2, ..., $n$ and add an edge $(x, y)$ iff $x < y$. This graph has the appropriate number of edges and cannot contain a cycle (any path visits an increasing sequence of nodes).

5. Assume that you have a network where every arc has a different capacity. Can this network have two different minimum cuts? Either give an example with two different minimum cuts or show that it cannot occur.

The graph below has two minimum cuts with value 3. The two cuts are $S_1 = \{s\}, T_1 = \{a, b, t\}$ and $S_2 = \{s, a, b\}, T_2 = \{t\}$.

6. In the network below, assume that s is the source and t is the sink. Find the maximum flow (both the value and the flow through each edge) and the minimum cut.
A maximum flow of 13 units is depicted below (some intermediate work should have been shown). One minimum cut is $S = \{s, c\}$ and $T = \{a, b, d, e, f, t\}$.

7. problem 1.4.4 (page 14)

a) Let $G$ be an acyclic directed graph. Select any node and repeatedly trace back along an incoming edge. Since the number of nodes is finite this process will either reach a vertex which has already been visited or you will reach a node with no incoming edges. The former is impossible, since you would have traced (in reverse) a cycle.

b) Let $G$ be an acyclic directed graph. For $i = 1, 2, \ldots, n$ find a source, label it $i$ and remove this vertex and all incident edges. This process creates the desired numbering.

If a graph $G$ has such a numbering then it must be acyclic, since any cycle must contain an edge from a higher numbered vertex to a lower numbered vertex.

8. problem 1.4.5 (page 14)

Assume you have a bipartite graph. From the definition you can label each vertex with either A or B such that there is no edge between two nodes with the same labels. Consider any cycle. The nodes in this cycle must alternate AB AB AB AB A or BA BA BA BA... BA B (where the first and last vertices are the same). Since you can pair up consecutive nodes (as above), the length of the cycle must be even.

Assume that you have a graph without odd cycles. For each connected component: choose any node $nd$. For any node an even distance from $nd$ label it A and otherwise B. This splits all nodes into 2 sets. There cannot be an edge between two nodes $nd1$ and $nd2$ both labeled A (or B) because there would be paths from $nd$ to $nd1$ and $nd2$ both of even (odd) length which combined with the edge from $nd1$ to $nd2$ would form an odd length cycle.