1. Alice wants to throw a party and she is trying to decide who to invite. She has n people to choose from, and she knows which pairs of these people know each other. She wants to invite as many people as possible, subject to two constraints:

   - For each guest, there should be at least five other guests that they already know.
   - For each guest, there should be at least five other guests that they don't already know.

   Describe and analyze an algorithm that computes the largest possible number of guests Alice can invite, given a list of n people and the list of pairs who know each other.

   (12 marks)

2. For any weighted graph $G = (V, E)$ and integer k, define G_k to be the graph that results from removing every edge in G having weight k or larger. Given a connected undirected weighted graph $G = (V, E)$, where every edge has a unique integer weight, describe an $O(|E| \log |E|)$-time algorithm to determine the largest value of k such that G_k is not connected.

   (12 marks)

3. Given a sorted array of distinct integers $A[1, \ldots, n]$, you want to find out whether there is an index i for which $A[i] = i$. Describe and analyze an algorithm that runs in time $O(\log n)$.

   (12 marks)

4. Let $G = (V, E)$ be a connected directed graph with positive edge weights, let s and t be vertices of G, and let H be a subgraph of G obtained by deleting some edges. Suppose we want to reinsert exactly one edge from G back into H, so that the shortest path from s to t in the resulting graph is as short as possible. Describe and analyze an algorithm that chooses the best edge to reinsert, in $O(|E| \log |V|)$ time. **(12 marks)**

5. In this game you throw balls at a line of bottles, each labeled with a number (positive, negative, or zero). You can throw as many balls at the bottles as you like, and if a ball hits a bottle, it will fall and shatter on the ground. Each ball will either hit no bottle, exactly one bottle, or two bottles (but only when the two bottles were **adjacent in the original lineup**). If a ball hits two adjacent bottles, you receive a number of points equal to the product of the numbers on the bottles. Otherwise, if a ball hits zero or one bottle, you do not receive any points. For example, if the line of bottle labels is $(5, -3, -5, 1, 2, 9, -4)$, the maximum possible score is 33, by throwing two balls at bottle pairs $(-3, -5)$ and $(2, 9)$. Given a line of bottle labels, describe an efficient dynamic programming algorithm to maximize your score. **(12 marks)**

6. The basic rule for blood donation is the following. A person's own blood supply has certain *antigens* present (we can think of antigens as a kind of molecular signature); and a person cannot receive blood with a particular antigen if their own blood does not have this antigen present. Concretely, this principle underpins the division of blood into four *types*: A, B, AB, and O. Blood of type A has the A antigen, blood of type B has the B antigen, blood of type AB has both, and blood of type O has neither. Thus,

patients with type A can receive only blood types A or O in a transfusion, patients with type B can receive only B or O, patients with type O can receive only O, and patients with type AB can receive any of the four types.

Consider the problem faced by a hospital that is trying to evaluate whether its blood supply is sufficient. Let s_O, s_A, s_B, and s_{AB} denote the supply in whole units of the different blood types on hand. Assume that the hospital knows the projected demand for each blood type d_O, d_A, d_B, and d_{AB} for the coming week. Describe and analyze a polynomial-time algorithm to evaluate if the blood on hand would suffice for the projected need. **(12 marks)**

7. Consider an arbitrary directed network $G = (V, E)$ with positive weights $w : E \to \mathbb{N}$ on edges and two distinguished vertices s and t.

   For any two vertices u and v, let $bottleneck_G(u, v)$ denote the maximum, over all paths π from u to v in G, of the minimum-weight edge along π.

   Describe and analyze an algorithm to compute $bottleneck_G(s, t)$ and a s-t path that achieves this bottleneck in $O(|E| \log |V|)$ time. **(12 marks)**

8. A *vertex cover* of an undirected graph $G = (V, E)$ is a subset of the vertices which touches every edge–that is, a subset $S \subset V$ such that for each edge $(u, v) \in E$, one or both of u, v are in S. The size of the vertex cover is given by $|S|$.

   Given a bipartite graph $G = (X \cup Y, E)$ as input, describe and analyze an algorithm to find a vertex cover of minimum size in G. For full credit your algorithm should run in time $O(|E||V|)$. **(15 marks)**