1. Suppose we are setting up ten workstations in a computer lab. We want to connect every workstation to a wired network in such a way that all of the networking ports on every workstation are used. Out of the ten workstations, one workstation has 6 networking ports, three workstations have 5 networking ports, four workstations have 4 networking ports, and two workstations have 3 networking ports. Prove or disprove: it is possible to connect every workstation in such a way that all of the networking ports on every workstation are used.

**Hint.** Consider the workstations to be vertices in a graph, and consider a workstation’s networking ports to be undirected edges incident to the vertex corresponding to that workstation.

**Solution:** Following the hint, we observe that this question is essentially asking us to find a graph on ten vertices with degree sequence \(6, 5, 5, 4, 4, 4, 4, 3, 3\).

If such a graph existed, then the sum of degrees of all vertices would equal 43. However, by the handshake lemma, we know that the sum of degrees of all vertices must be twice the number of edges in the graph; that is, the sum of degrees must be an even number. Since 43 is an odd number, no such graph exists.

2. An \(n\)-bit Gray code is a set of bit strings of length \(n\) that uses a special incrementation method. Instead of the usual binary incrementation method, which increments by flipping one or more bits at a time (i.e., \(000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow \cdots\)), a Gray code increments by flipping exactly one bit at a time. As an example, the following sequence lists the elements of a 2-bit Gray code:

\[
00 \rightarrow 01 \rightarrow 11 \rightarrow 10.
\]

(a) Illustrate a 3-bit Gray code. Begin by drawing a graph where each vertex corresponds to a bit string of length 3 and where an edge \(\{u, v\}\) exists when you can obtain the bit string at vertex \(v\) by flipping one bit in the bit string at vertex \(u\). Next, find a Hamiltonian circuit through your graph that corresponds to the sequence of single bit-flips needed to generate your Gray code.

**Solution:** The following graph contains all vertices corresponding to bit strings of length 3 and all edges joining bit strings that differ in one bit.

![Graph](image)

The Hamiltonian circuit depicted in the graph corresponds to the following 3-bit Gray code:

\[
000 \rightarrow 001 \rightarrow 011 \rightarrow 010 \rightarrow 110 \rightarrow 111 \rightarrow 101 \rightarrow 100.
\]

(b) Name an application where we might prefer to use an \(n\)-bit Gray code over the usual binary numbers. Why would we prefer Gray codes in that case?
**Solution:** There are a large number of applications where Gray codes provide a benefit over the usual binary numbers. Some common examples include error correction in data communications, switch position encoding, Boolean circuit minimization, and solving the Tower of Hanoi puzzle. (This list is by no means exhaustive.)

3. **[5 marks]** Suppose you host a study session and you invite six classmates. Given any pair of people not including yourself, those people may be friends (meaning they know each other) or strangers (meaning they do not know each other).

We can illustrate the property of people being friends or strangers by constructing a graph edge colouring. Given a graph with six vertices, we colour an edge between two vertices red if the people represented by those vertices are friends, or blue if the people are strangers.

Using the notion of graph edge colouring, prove that either at least three people at your study session are pairwise friends or at least three people at your study session are pairwise strangers.

*(You should give a general proof. Do not simply draw an example of a graph edge colouring.)*

**Solution:** Begin by constructing the complete graph $K_6$. This graph contains one vertex for each person and one edge from every person to every other person. An edge denotes the relationship between two people.

Choose an arbitrary vertex $u$. There are five edges leading from $u$ to each of the other vertices of the graph. Using the notion of graph edge colouring, each of these edges may be coloured either red or blue. By the pigeonhole principle, at least three of these five edges must be the same colour. Without loss of generality, assume these three edges are coloured red.

Suppose that the three red edges lead to the vertices $v_1, v_2$, and $v_3$. If any of the edges $\{v_1, v_2\}, \{v_2, v_3\}$, or $\{v_1, v_3\}$ are red, then that edge together with two edges from $u$ form a pairwise relationship between three people. Otherwise, if none of the specified edges are red, then those three edges also form a pairwise relationship between three people. In any case, we obtain the desired result.

4. **[5 marks]** You have been selected to judge the annual discrete mathematics tournament, where students compete against one another to answer challenging questions and win big prizes.

The rules of the tournament state that all students compete against all other students and, for each match between two students, exactly one of the students wins the match.

With your knowledge of graph theory, you decide to record the outcomes of the tournament in the following way: you create a graph where each student is a vertex and where, given vertices $u$ and $v$, exactly one directed edge $\{u, v\}$ or $\{v, u\}$ exists in the graph.

If $n$ students compete in the tournament, how many possible “outcome graphs” exist?

*Hint.* To determine this answer, you must count both the number of ways to choose pairs of students from the set of $n$ students and the outcome of each individual match.

**Solution:** We begin by modelling the set of students as a graph containing $n$ vertices. For each match, we choose two vertices and assign an edge between them. Since order does not matter and repetition is allowed, we have a total of $C(n, 2) = \frac{n \times (n-1)}{2}$ ways to make these choices.

For each pair of vertices, we have two possibilities for the direction of the edge connecting those two vertices in the graph. In this case, order does matter and repetition is allowed, so we have a total of $2^{(n \times (n-1))/2}$ ways to make these choices.

Altogether, a total of $2^{(n \times (n-1))/2}$ “outcome graphs” exist.