1. (1.3.14) Show that every simple graph with at least two vertices has two vertices of equal degree.

2. (1.3.57) Let \( d_1 \geq d_2 \geq \cdots \geq d_n \geq 0 \). Show that if \( d_1 < n \), \( d_1 \leq d_n + 1 \) and \( \sum_i d_i \) is even then \( d_i \) is a graphic sequence.

3. A parallel computer uses a large number of processors in concert. Let \( G \) denote the graph whose vertices are the processors and whose edges represent direct communication links. We’d like \( G \) to have very few edges because too many connections are unwieldy, yet we’d like to have \( G \) be “well-connected”. One good example of such a graph is the \( k \)-dimensional hypercube, \( Q_k \) (see section 1.3.7 and the following in the book). It has \( n = 2^k \) vertices and \( (n \log_2(n))/2 = k2^{k-1} \) edges.

   (a) (Vertices in \( Q_k \) are connected by very short paths.) What is \( \text{diam}(Q_k) \)?

   (b) (Vertices in \( Q_k \) are connected by lots of paths.) Given two vertices \( v \) and \( w \) of \( Q_k \), give a formula for the number of distinct shortest paths from \( v \) to \( w \). (Hint: The number will depend on \( d(v, w) \)).

   (c) (A vertex in \( Q_k \) is connected by short paths to lots of vertices.) Given a vertex \( v \), how many vertices are at distance \( i \) from \( v \)?

   (d) Let \( B_k \) denote the graph whose vertices are the subsets of \( \{1, \ldots, k\} \), where two sets \( A, B \) are connected by an edge if and only if \( |A \triangle B| = 1 \). Show that \( B_k \cong Q_k \).

4. Show that if \( G \) is an connected Eulerian graph and \( T \) is a trail such that \( G - E(T) \) is connected, then \( T \) can be extended to an Eulerian circuit. (\( G \) has an Eulerian circuit \( C \) that has \( T \) as an initial segment.)

5. Show that when given a loopless graph \( G \) on the vertex set \( V(G) = \{1, 2, 3, \ldots, n\} \) as input the following algorithm produces a bipartite subgraph \( B \) having at least \( |E(G)|/2 \) edges.

   Given \( G \), let \( X = \{1\}, Y = \emptyset \). For \( i = 2 \) to \( n \), if \( i \) is adjacent to more edges with endpoints in \( X \) than to edges with endpoints in \( Y \) add \( i \) to \( Y \). Otherwise, add \( i \) to \( X \). Output the subgraph \( B \) whose edges are the edges of \( G \) that have one endpoint in \( X \) and one endpoint in \( Y \).