Problem 4:
The maximum degree is 3, since you can either flip the last bit, or shift in one direction, or shift in the other direction; some vertices may have smaller degree since the shifts may be symmetric. The diameter is 2d − 1; any vertex \( v = (v_{d-1}, v_{d-2}, \ldots, v_1, v_0) \) can reach any other vertex \( w = (w_{d-1}, w_{d-2}, \ldots, w_1, w_0) \) in 2d − 1 edges by the following procedure: flip \( v_0 \) if it is not equal to \( w_0 \). Then shift to the right, and compare \( v_1 \) (which is now in the last bit position) to \( w_2 \), flipping if necessary. Repeat this sequence, until after \((d−1)\) possible flip/shift combinations, we have the original \( v_{d−1} \) in the last bit position, which we then flip if it is not equal to \( w_0 \). This produces the coordinates of the vertex \( w \), using 2\((d−1)\) + 1 = 2d−1 operations. It is in fact necessary to perform 2d−1 operations to move between vertex \((0,0,\ldots,0)\) and \((1,1,\ldots,1)\).

Problem 5:
The DeBruijn graph has a max degree of 2b since each vertex is connected to those vertices which share its first \((d−1)\) bits as their last \((d−1)\) bits but with each possible first bit (therefore the \(b\)), and also those vertices which share its last \((d−1)\) bits as their first \((d−1)\) bit but with every possible last bit (therefore the second \(b\)). No other connections are possible to that vertex, and therefore the maximum degree is 2b. The largest distance between any two nodes is \(d\), because with each edge shift you can get a new bit - therefore after \(d\) edge shifts you can replace the entire original address with the new one even if there’s no correlation.

One can express the number of nodes in the graph as \(n = b^d\), since each of the \(d\) digits can take on \(b\) different values. Replace \(b\) with \(\delta/2\), for \(\delta\) being the maximum degree. Then invert to get \(d = \frac{\log \delta}{\log \delta - 1} \approx \frac{\log n}{\log \delta - 1}\). This then implies that the DeBruijn graph is an infinite family of graphs with max degree \(\delta\) and \(n\) vertices with a diameter equal to \(\frac{\log n}{\log \delta - 1}\), which proves Theorem 2.10 by construction.

Problem 6:
For even \(n\), split the torus into a left and right half (for one of the two digits, let \(U\) contain all vertices with the digit in the lower half of its range). Each half has \(n^2/2\) vertices, and there are \(2n\) edges between the halves (the \(n\) edges crossing the middle and the \(n\) circular edges that distinguish the torus from the mesh). Then \(c(U, U) = 2n\). And since there are \(n^2/2\) nodes, and each has a degree of 4 (going up or down one in each digit), \(c(U) = c(U) = 4 \cdot n^2/2 = 2n^2\).

Then \(\frac{c(U, U)}{\min(c(U), c(U))} = \frac{4n}{n} = 4\).

For odd \(n\), let \(U\) contain \(\frac{2n−1}{2}\) columns of vertices and half of one of the adjacent columns. Then the number of vertices is \(\frac{n-1}{2} + \frac{n-1}{2} = n^2/2\), so \(c(U) = 2n^2 – 2\). Now \(c(U, U) = 2n + 2\) since the additional half-column creates two additional outgoing edges, up and down to the rest of the column. Then the edge expansion is \(\frac{2n+2}{\frac{2n^2−2}{n}} = \frac{1}{n−1}\).

Problem 7:
Any \(d\)-dimensional hypercube can be broken down into two \((d−1)\)-dimensional hypercubes (for example, by taking all vertices with the first bit 0 as one hypercube, and all vertices with the first bit 1 as the other). Take one of these hypercubes to be \(U\). Then \(c(U, U) = 2^{(d−1)}\) (assuming all edges have capacity 1) since you connect each pair of vertices which are identical in the other \((d−1)\) positions. But \(c(U) = c(U) = d\cdot 2^{(d−1)}\) since each of the vertices in one hypercube has \(d\) edges. Then:

\[
\frac{c(U, U)}{\min(c(U), c(U))} = \frac{2^{(d−1)}}{d^2(2^{(d−1)})} = \frac{1}{d}
\]