Assignment 3

1. (10%) What is the least number of links that must fail in a hypercube of order n to prevent us from guaranteeing that there is a path between two arbitrary nodes in the hypercube? Justify your answer.

2. (10%) What is the least number of links that must fail in a completely-connected network of order n to prevent us from guaranteeing that there is a path between two arbitrary nodes in the hypercube? Justify your answer.

3. (20%) Consider the following portions of two different programs running at the same time on two processors in a shared memory multiprocessor (SMP). Assume that before the code is run, x = 3 and y = 5.

   ```
   Processor 1: ...; x = x - 3; y = y * 2 ; ...
   Processor 2: ...; x = x * y; y = x ; ...
   ```

   What are all of the possible resulting values of x and y assuming that the code is implemented using a load-store architecture? For each possible outcome, show an interleaving that results in that value. (You have to consider the assembly language sequences.)

4. (20%) Consider a 2-dimensional mesh with n^2 nodes.

   (a) The shortest distance that a message travels in this mesh is 1 hop; the longest distance is $2(n-1)$ hops. If all possible destination nodes are equally likely, what is the average distance a message needs to travel from a corner node to any other node? You must prove your result.

   (b) Is it possible to partition this mesh into two disjoint subsets, A and B, such no node in A is directly connected to any other node in A, and no node in B is directly connected to any other node in B, and every edge from any node connects to a node in the other set? If so, describe the sets, and if not, explain why.

5. (10%) How many nodes are at a distance of exactly k edges from a source node s in a d-dimensional hypercube, where $0 \leq k \leq d$? Probe that your answer is correct.

6. (30%) Suppose that an array A of size 2^n integers is in sorted order. Given a number key, we could use binary search to find key in the array, or return -1 if it is not found. Suppose that we could use a SMP with p processors, with p much smaller than 2^n, to search for the key. Using a strategy similar to the one for summing an array, and assuming that the barrier synchronization function `synch()` is available on the machine, write the algorithm. Try to design it so that it is as efficient as possible.