Homework 1

Xinyuan Miao
020033910009
mxinyuan@sjtu.edu.cn

Exercise 2.5

We Encode each node in binary and the distance of each 2 nodes is the Hamming distance that equals to the number of bits with different values in the code.

Proposition 1. For each 2 nodes which are k units apart in a hypercube, there are at most k paths of length k between the 2 nodes that share no edges.

Proof. The path that satisfies the condition must continue along the direction that reduces the hamming distance between the two nodes, which equals to change 1 bit for each step and finally get the source node's code the same as that of the destination node. Note that for the first step, there are k different choices to make since there are total k bits with different values, which implicates the number of the path satisfying the condition will not be larger than k. □

Proposition 2. For each 2 nodes which are k units apart in a hypercube, there are k paths of length k sharing no edges between the 2 nodes.

Proof. We consider the k bits with different values in the code of the 2 nodes and let the length-k code of the source node $s = (0, 0, \ldots, 0)$ and the destinatnio node $t = (1, 1, \ldots, 1)$ without the loss of generality. We consider the following paths:

$P_1 = (0, 0, 0, \ldots 0) \to (1, 0, 0, \ldots 0) \to (1, 1, 0, \ldots 0) \to \cdots \to (1, 1, 1, \ldots 1, 0) \to (1, 1, \ldots 1, 1)$
$P_2 = (0, 0, 0, \ldots 0) \to (0, 1, 0, \ldots 0) \to (0, 1, 1, \ldots 0) \to \cdots \to (0, 1, 1, \ldots 1, 0) \to (1, 1, \ldots 1, 1)$
$\ldots$
$P_k = (0, 0, 0, \ldots 0) \to (0, 0, 0, \ldots 1) \to (1, 0, 0, \ldots 1) \to \cdots \to (1, 1, 1, \ldots 0, 1) \to (1, 1, \ldots 1, 1)$
And we get k different paths satisfying the condition. □

Exercise 2.11

1. $\frac{2}{n} \log_2 n$ switching elements
2. Diameter is $\log_2 n - 1$
3. Bisection width is n
4. Maximum number of edges per switching node is 4
5. As the number of nodes increase, the longest edge length is constant

Exercise 2.14

10 millions ops/sec $\Longleftrightarrow$ 0.1 μsec/ops

$$Performance = \frac{\text{Size operations}}{\lceil \frac{\text{Size}}{8} \rceil \times 0.1 \mu sec}$$

When Size = 1

$$Performance = 10 \text{ millions ops/sec}$$

When Size = 50

$$Performance = \frac{50 \text{ operations}}{0.7 \mu sec} = 71.43 \text{million ops/sec}$$

Exercise 2.18

1. To prevent accesses to the cache directory from becoming a performance bottleneck.
2. Avoid the consistency problem of the directory content.