HW9:

Huffman Coding: Huffman Coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code, and the least frequent character gets the largest code.

HW10:

The Floyd Warshall Algorithm is for solving the All-Pairs Shortest Path problem. The problem is to find shortest distances between every pair of vertices in a given edge weighted directed Graph. We initialize the solution matrix same as the input graph matrix as a first step. Then we update the solution matrix by considering all vertices as an intermediate vertex. The idea is to one by one pick all vertices and updates all shortest paths which include the picked vertex as an intermediate vertex in the shortest path. When we pick vertex number k as an intermediate vertex, we already have considered vertices {0, 1, 2, .. k – 1} as intermediate vertices. For every pair (I, j) of the source and destination vertices respectively, there are two possible cases.

1. K is not an intermediate vertex in shortest path from I to j. We keep the value of dist[i][j] as it is.
2. K is an intermediate vertex in shortest path from I to j. We update the value of dist[i][j] as dis[i][k] + dis[k][j] if dis [i][j] > dis[i][k] + dis[k][j]

HW11:

1. For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle if CLK is high, which makes the output of the flip-flop unstable or uncertain. This problem is called race around condition in J-K flip-flop. Race Around Condition can be avoided by ensuring that the clock input is at logic “1” only for a very short time.
2. When more than one processes access a same code segment that segment is known as critical section. Critical section contains shared variables or resources which are needed to be synchronized to maintain consistency of data variable. For example, a thread must acquire a lock prior to executing a critical section. The lock can be acquired by only one thread.
3. Mutual exclusion is a property of process synchronization which states that “no two processes can exist in the critical section at any given point of time”. Any process synchronization technique being used must satisfy the property of mutual exclusion, without which it would not be possible to get rid of a race condition. For example, in the clothes section of a supermarket, two people are shopping for clothes. Boy A decides upon some clothes to buy and heads to the changing room to try them out. While boy A is inside the changing room, there is an ‘occupied’ sign on it – indicating that no one else can come in. Girl B must use the changing room too, so she must wait till boy A is done using the changing room. Once boy A comes out of the changing room, the sign on it changes from ‘occupied’ to ‘vacant’ – indicating that another person can use it. Hence, girl B proceeds to use the changing room, while the sign displays ‘occupied’ again. The changing room is nothing but the critical section, boy A and boy B are two different processes, while the sign outside the changing room indicates the process synchronization mechanism being used.