**CS6550 Advanced Operating Systems**

**Homework Assignment 1**

**Question 1:**

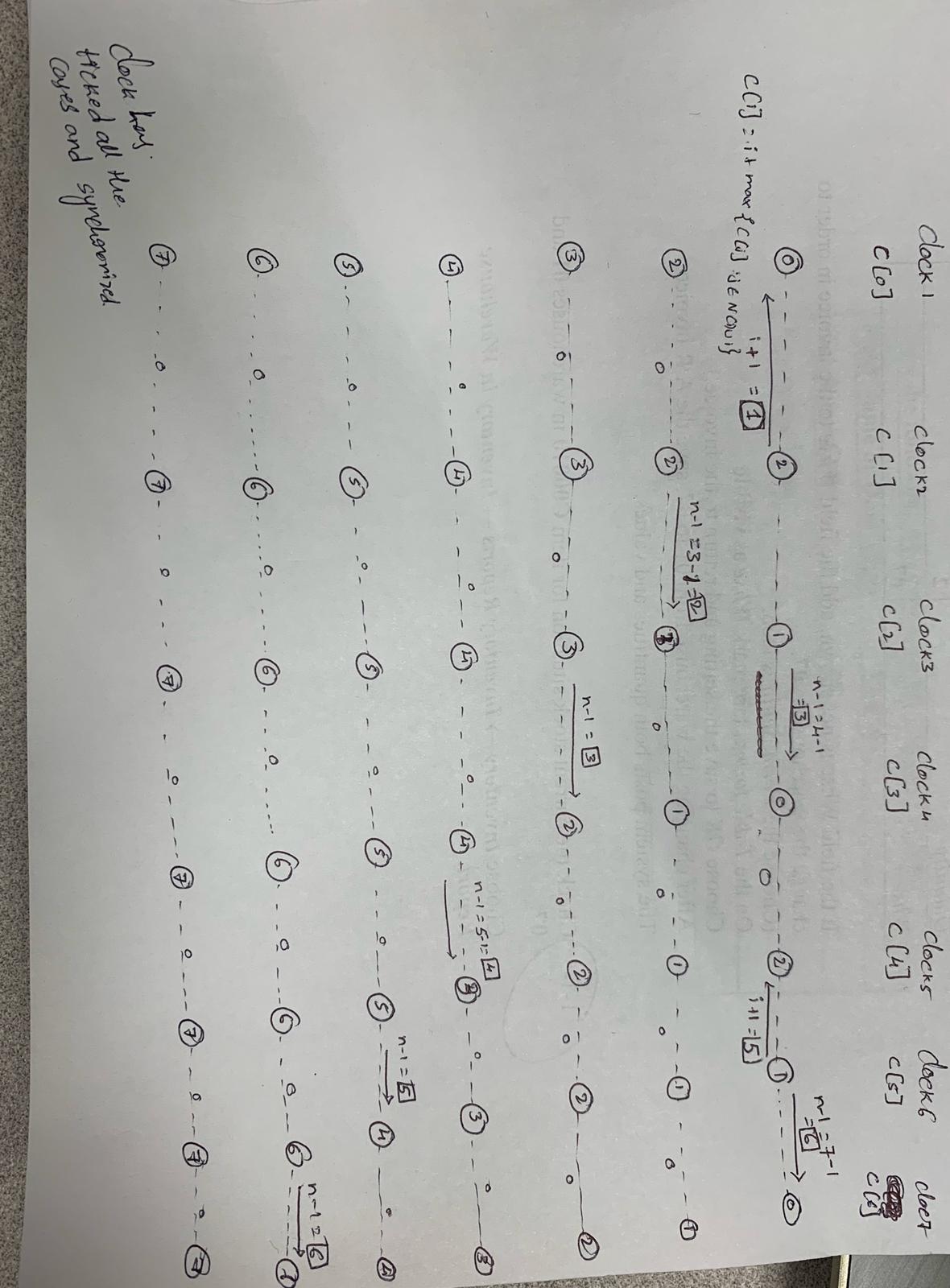
We are assuming the synchronous model where the clock executes the action c[i], N(i) with a tick. Our aim is to prove the given program for clock is synchronized and it’s in synchronize phase always.

C[i]: It represent value/phase of clock i.

N(i): It represent the neighbors of clock i.

In this we use variant function to prove the clock is synchronized. We use shared variable model. Here each clock reads the set of neighbors and update the clock and continue its process. In this we are going to tick the clock i until the clock J has equal number of value of neighbors of clock i. In this Clock we have two-way communication so we are not going to lose any ticks in the clock each tick is counted. In clock synchronization we have two algorithms Berkeley algorithm and Cristian’s algorithm. In this Program if we have any electrical disturbance, we use rate correction just by speed up/ slow down local tick rate. Rate of time is slightly incorrect for a while. If we are using this clock synchronization on internet, we use Network Time Protocol to get good accuracy.

**Example:**

****

**1st tick:**

Clock[0]: ( 1 + max of its neighbors) 1+0= 1 is the time for clock[0]

Clock[1]: (1 + max of its neighbors) 1+1= 2 is the time for clock[1]

**2nd tick:**

Clock[2]: (1 + max of its neighbors) 1+2= 3 is the time for clock[2]

Clock[3]: (1 + max of its neighbors) 1+3= 4 is the time for clock[3]

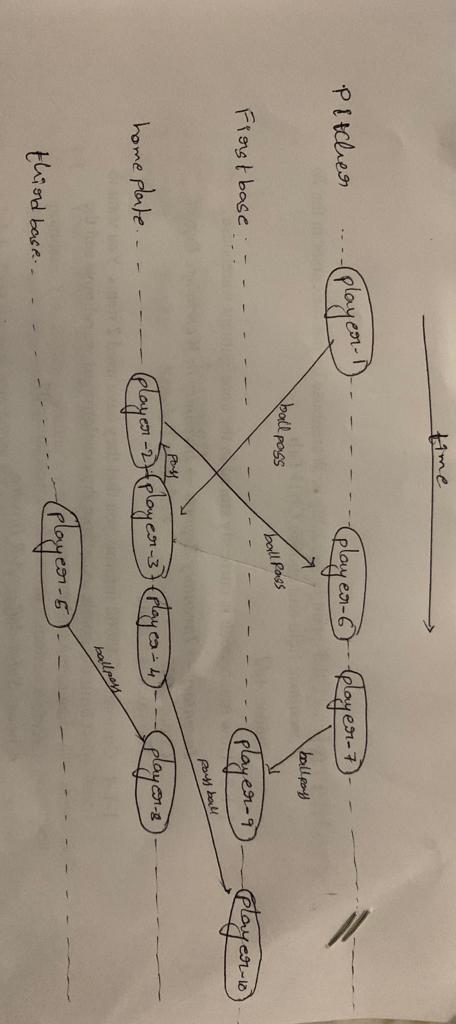
Clock[4]: (1 + max of its neighbors) 1+4= 5 is the time for clock[4]

Clock[5]: (1 + max of its neighbors) 1+5= 6 is the time for clock[5]

We can also access the clock by using the neighborN(i) -1 as shown in the figure.

**Example:** Base-ball game.

* In the game Pitcher knows player-1 happens before player-6, which happens before player-7
* The Home plate umpire knows player-2 is before player-3, which is before player-4, which is before player-8, which is before player-10
* Player-8 and Player-9 has unclear relationship.



The clock is synchronous even after there is an clock changes for electrical disturbances. The each clock execute the tick in same amount of time, from this we can tell the process take same number of steps to synchronize.

**Question 2:**

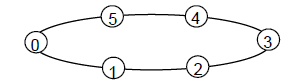
Here we have 6 process the system is asynchronous. In this Process execute the program in an asynchronous process each and every process

P(i) : integer which is initialized to 0.

N(i) : it’s the neighbor of process i.

In this process if the executing process of P(j) is less than the P(i) initial variable then process skips.

The second step If all neighbors of P(i) is less than P(j)+1 gives to P(i) then we increment P(i) process and execute all the process.



The process a the concluding state is { p0, p1, p5, p4, p2, p5, p1, p0}

During an execution, what is the maximum possible difference between the values of *p* of two different processes is 3.

Starting from the initial state(p0, p1, p5, p2, p4, p3), present the steps that via which the difference between two *p’*s can reach the maximum value. The process starts from 0 and increase to 3 from the range between p0 and p3.

**Question 3:**

The