C1 Review Test (Feb 2022)

Total Marks: 30
Time: 60 minutes

Instructions:

- Write your Name, Roll Number and Mobile Number on the top right corner of the first page.
- Please answer based on your understanding of the question. Clearly specify the assumptions made, if any.
- Give proper, precise reasoning for each answer.

1. Is the following code parallelizable? Justify your answer: If yes, show how it can be parallelized and if not, give proper reason.

   [3 marks]

```
#pragma omp parallel for
for (int i = 1; i < N; i++)
{
A[i] = B[i] - A[i - 1];
}
```

2. Consider a situation where you need to parallelize a given problem with 10 threads. If you divide the problem into 10 subproblems of equal size, you would expect a speedup of 10x. Compute the speedup using Amdahl's law if one of the subproblems is twice the size of others.

   [5 marks]

3. Consider the following distributed execution that uses logical clocks (see Figure 1). For a local state s on process P_i, let the logical timestamp of s be the timestamp of the closest preceding event. Thus for s, we define three clock values: lts(s) denotes lamport's logical clock value at s, $vts(s)$ denotes vector time at s and $mts(s)$ denotes matrix time at s.

   In this figure, squares represent local states and circles represent events.

   (a) Compute vts(a), vts(b), vts(c).

   [3 marks]

   (b) Compute mts(a), mts(b), mts(c).

   [3 marks]

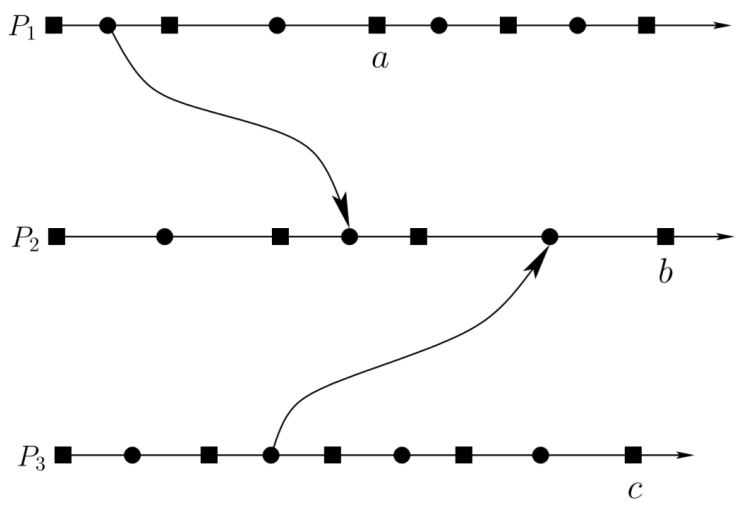

Figure 1: Pictorial representation of an execution

4. (a) Singhal–Kshemkalyani's differential technique works with which of the following type(s) of communication channels:
   i. FIFO
   ii. Non-FIFO
   iii. Both FIFO and Non-FIFO
   iv. None

   Justify your answer.

   [5 marks]

   (b) Prove or disprove: if the lamport's timestamps of every pair of local states in a global state G differ by at most one, then G is a consistent global state. Formally, $\langle \forall i, j :: (lts(G[i]) - lts(G[j])) \leq 1 \rangle \Rightarrow G$ is consistent

   [5 marks]

5. Consider Lamport and Melliar-Smith's algorithm for internal clock synchronization. This algorithm handles two-faced clocks. It guarantees that in a system of n processes, the clocks remain synchronized even if there are at most t two-faced clocks, when $n > 3t$. The algorithm assumes that the clocks are resynchronized often enough so that no two non-faulty process clocks ever differ by more than δ. As per this algorithm, the maximum difference between the averages computed by any two non-faulty processes is $3t\delta/n$. Please explain how.

   [6 marks]