1. Assume we have the following task graph. A task can be thought of as function/procedure. Every task is labeled with its run time on a core. An arrow from a task to another means that the first task generates data needed by the second one. Assume all data are of the same size.

a. How will you parallelize that program?
b. What will be the speedup if we have 2 processes? 4 processes? 8 processes?
c. What is the span for the above graph? what is the work?
d. What is the parallelism?
2. Why did we move from single core to multicore processors?

3. To what type of programs is shared-memory architecture more suitable? How about distributed-memory, which programs benefit from it?

4. How can we parallelize n! (factorial n) calculation using MPI?

5. When do we need to use derived data type in MPI?

6. According to Amdahl’s law, what is the maximum speed-up of a parallel computation given that 80% of the computation can be executed in parallel?

7. What is the difference between MPI_Reduce and MPI_Scan?

8. Why are critical sections not used in MPI programs?

9. When do you decide to split the communicator?

10. Can a serial program be faster than a parallel version? Explain?