Module 1 Quiz

Total points 10

1.

Question 1

Consider the following parallel pseudo-code that uses the async-finish notation for task creation and termination introduced in Lecture 1.1:

```
finish {
  async S1;
  finish {
    async S2;
    S3;
  }
  S4;
}
S5;
```

Which of the following statements are true? *Check all that apply*

1 point

A. S2 can potentially execute in parallel with S3

B. S2 can potentially execute in parallel with S4

C. S1 can potentially execute in parallel with S4

D. S1 can potentially execute in parallel with S5

2.

Question 2

Consider the following parallel pseudo-code that uses the async-finish notation for task creation and termination introduced in Lecture 1.1::

```
S1;
finish {async S2;}
S3;
```

The line "finish {async S2;}" could be equivalently replaced by which of the following? You may assume that S1, S2, and S3 do not spawn any nested async tasks.

Check all that apply

1 point

A. async S2;

B. async finish S2;

C. S2;

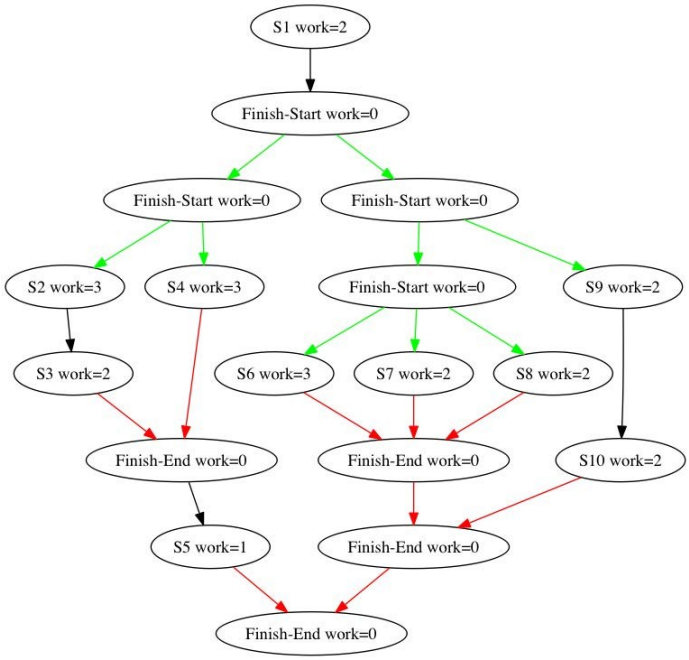
D. finish S2;

3.

Question 3

Consider the computation graph in Figure 1 below (where black, green and red arrows represent continue, fork, and join edges respectively, as explained in Lecture 1.3).

Figure 1.



What is the total WORK for the computation graph in Figure 1? Please enter an integer. **22**

1 point

4.

Question 4

What is the SPAN or CPL (critical path length) for the computation graph in Figure 1? Please enter an integer. **8**

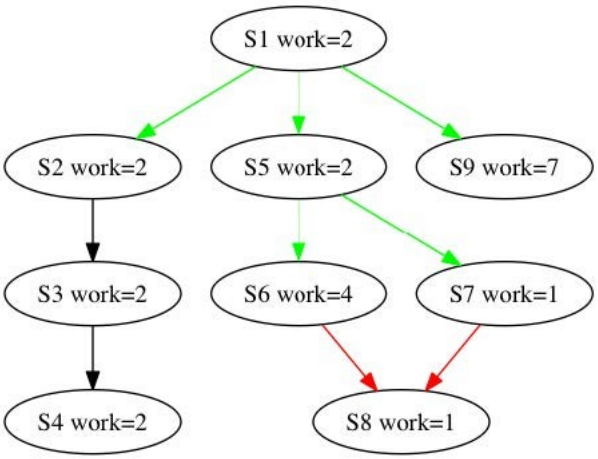
1 point

5.

Question 5

Consider the computation graph in Figure 2 below (where black, green and red arrows represent continue, fork, and join edges respectively, as explained in Lecture 1.3).

Figure 2.



For the computation graph in Figure 2, identify which of the following paths are critical paths?

Check all that apply

1 point

A. S1 → S2 → S3 → S4

B. S1 → S5 → S8

C. S1 → S5 → S6 → S8 (MOST WORK 9)

D. S1 → S5 → S7 → S8

E. S1 → S9 (TIED FOR MOST WORK 9)

6.
Question 6

Recall that the concepts of *WORK* and *SPAN* were introduced in Lecture 1.3. Consider the pseudo-code in Figure 3 below for adding two lower triangular matrices (square matrices in which all the elements above and including the (0,0) to (n,n) diagonal are zero), in which each execution of the statement A[i][j] = B[i][j] + C[i][j]; represents one unit of work in the computation graph, WORK(S5) = 1:

Figure 3.

```
finish {
  for (int i = 0; i < n; i++) {
    async {
      for (int j = 0; j < i; j++) {
        A[i][j] = B[i][j] + C[i][j];
```

```
    } // for-j
  } // async
 } // for-i
}
```

The total WORK performed by the program in Figure 3 (after it completes execution), in terms of n equals _____.

1 point

A. 1

B. n−1

C. n(n−1)/2

D. None of the above

7.

Question 7

The Critical Path Length (CPL) of the program in Figure 3 in terms of n equals _____.

1 point

A. 1

B. n−1

C. 2n(n−1)

D. None of the above

8.

Question 8

Recall that Ideal Parallelism was also defined in Lecture 1.3. The Ideal Parallelism of the program in Figure 3, as a function of n equals _____.

Figure 3

```
finish {
 for (int i = 0; i < n; i++) {
  async {
   for (int j = 0; j < i; j++) {
    A[i][j] = B[i][j] + C[i][j];
   } // for-j
  } // async
```

} // for-i

} // finish

1 point

A. 1

B. n/2

C. n−1

D. None of the above

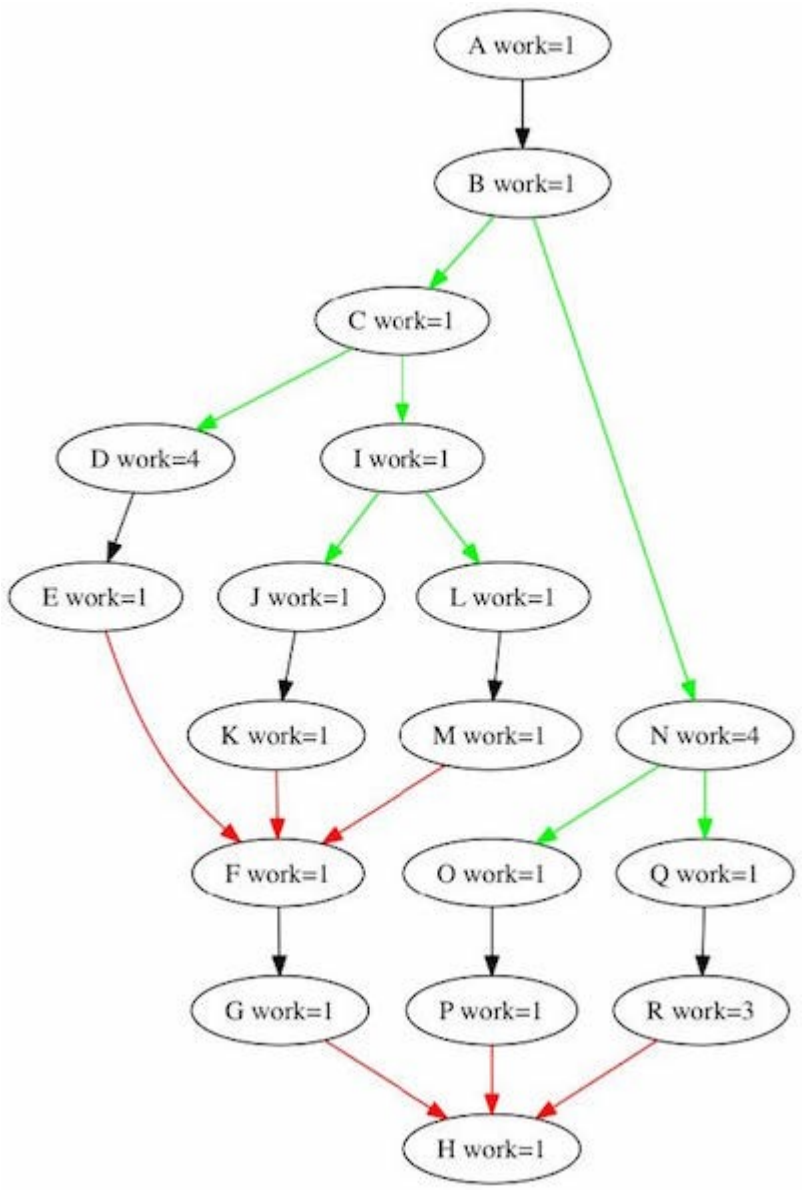
9.

Question 9

Recall that multiprocessor schedules were introduced in Lecture 1.4. Though, by default, we focus on schedules with no unforced idleness in this course, this question will allow for all possible legal schedules, including those that may have unforced idleness, i.e., schedules in which a process may be idle even if there is work that is available to be executed.

Consider the computation graph shown below in Figure 4. Select the statement below that is true about legal schedules for this graph on 3 processors.

Figure 4.



1 point

A. There exists a legal schedule in which node C can start execution before node B starts.

B. There exists a legal schedule in which node P can complete execution before node C starts. (ONLY ONE DUE TO WORK COUNTS AND PRECEDENCE)

C. There exists a legal schedule in which nodes J, L, O, Q can all be scheduled at the same time.

10.

Question 10

For an arbitrary computation graph and its schedule on P processors, which of the following relationships must always hold between Speedup(P) and Ideal Parallelism?

1 point

A. Speedup(P) < Ideal Parallelism

B. Speedup(P) ≤ Ideal Parallelism

C. Speedup(P) = Ideal Parallelism

D. Speedup(P) ≥ Ideal Parallelism

E. Speedup(P) > Ideal Parallelism