

Take a note of your solutions for part 2 below.

Part 1: Solving.

Demo:

Nothing depends on anything else. Remember, we can have only 4 counters in a row!

Scenario 1:

A->B; B->C; C->D; D->E; F->G; G->H; H->I; I->J

Scenario 2:

A -> [B, C, D, E]; E->F; F -> [G, H, I, J]; (All of B, C, D and E depend on A etc)

Scenario 3:

[A, B, C] -> D; D -> [E, F, G]; G -> [H, I, J]; (D depends on all of A, B and C etc)

Scenario 4:

A -> [B, C, D, E, F, G, H, I]; I->J;

Part 2: Assessing.

How many rows does each scenario need to complete? What is the minimum possible for all 10 using 4 processors?

You can easily work out the 'average' load, assuming each task takes the same amount of time. What are the average loads for your solutions above?

Now suppose that task A takes twice as long (2 rows) to complete as the rest. How does this affect the average load and the time-to-solution?

Suppose instead that you have 6 processors (maximum 6 counters in a row). Which scenarios can be completed faster?