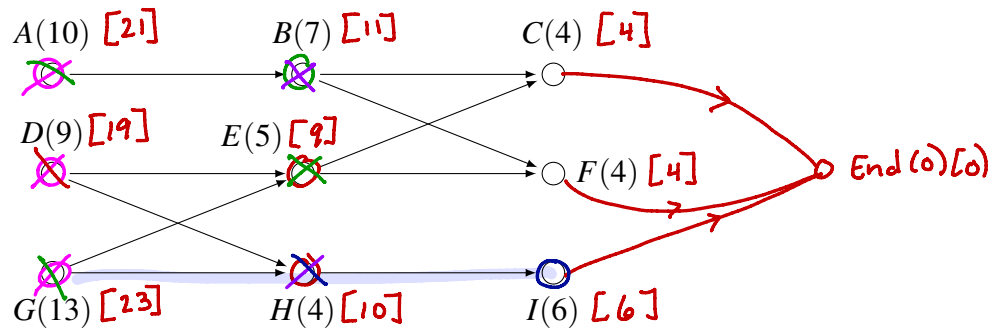


Worksheet Scheduling 2: Critical Path Algorithm

1. Consider the digraph from the previous scheduling worksheet:

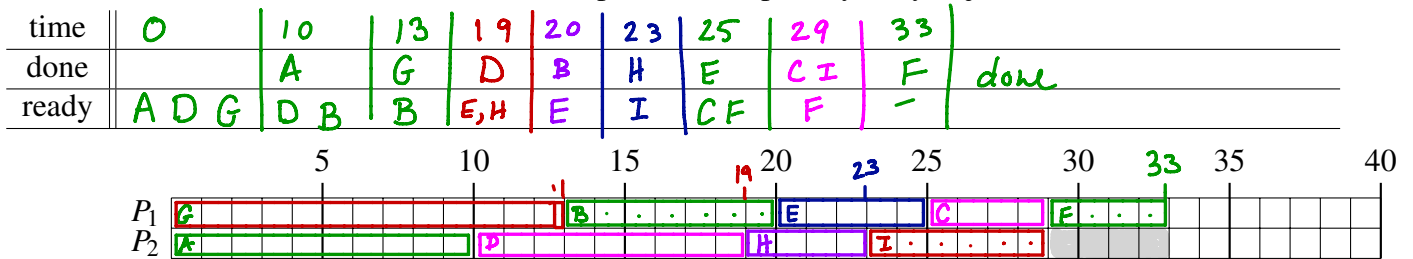


✓ (a) Use the backflow algorithm to label each vertex in the digraph.

(b) Construct a priority list using the Critical Path algorithm.

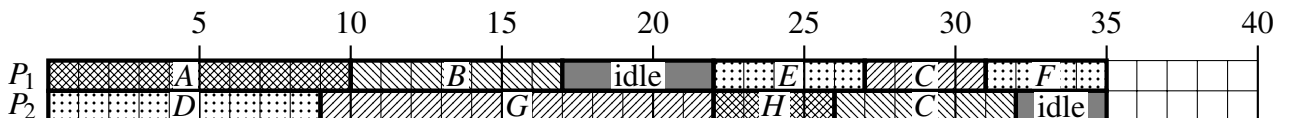
G A D B H E I C F
23 21 19 11 10 9 6 4 4

(c) Construct a schedule that corresponds to the priority list you just found.

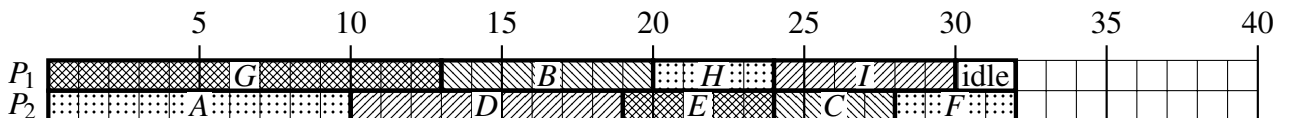


(d) The schedules that you found on the previous worksheet are shown below:

Priority List A, B, C, D, E, F, G, H, I



Priority list from the Decreasing Time Algorithm: G, A, D, B, I, E, C, F, H



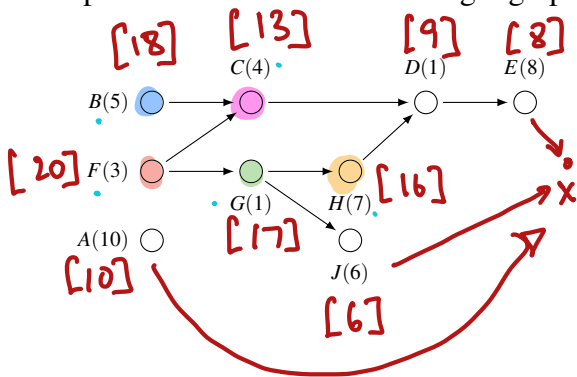
(e) How can you identify the overall critical path given the labels you put on the digraph from the backflow algorithm?

Start with the vertex with largest label. Go to neighbor of largest label.

(f) How does the schedule you found using the critical path priority list compare to the other schedules you found?

It's better than alphabetical order. A little worse than decreasing time algorithm.

2. Typically the Critical Path algorithm produces a very good schedule, but it may or may not be optimal. Consider the following digraph:



(a) Construct the priority list corresponding to the decreasing time algorithm.

A, E, H, J, B, C, F, D, G

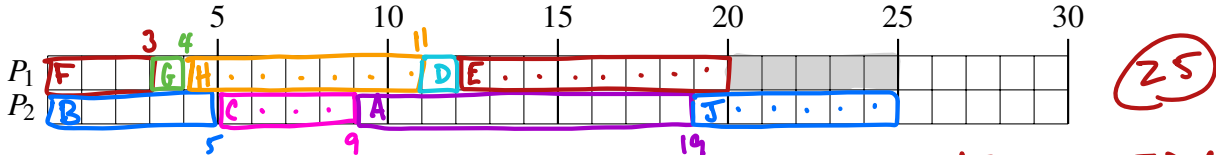
(b) Label the graph using the backflow algorithm.

(c) Construct the priority list corresponding to the critical path algorithm.

F, B, G, H, C, A, D, E, J

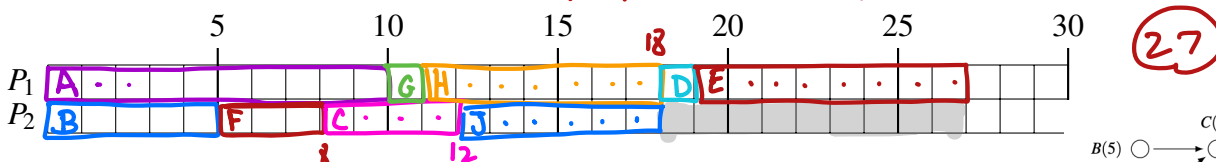
(d) Construct the schedule using the **critical path** priority list.

time	0	3	4	5	9	11	12	19	20	25
done		F	G	B	C	H	D	A	E	J
ready	F, B, A	G, A	H, J	C, A, J	A, J	D, J	E, J	J		



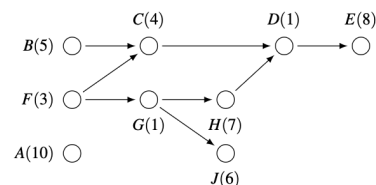
(e) Construct the schedule using the **decreasing time** priority list.

time	0	5	8	10	11	12	18	19	27
done		B	F	A	G	C	H, J	D	E
ready	A, B, F	F	C, G	G	H, J	J, D	D	E	

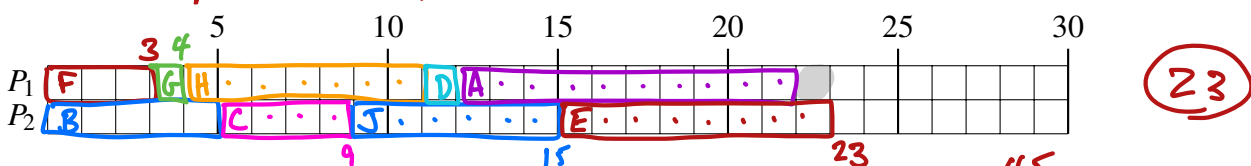


(f) Construct a schedule using the priority list

F, B, G, C, H, J, D, A, E



time	0	3	4	5	9	11	12	15
done		F	G	B	C	H	D	E
ready	F, B, A	G, A	H, J, A	C, J, A	J, A	D, A	A, E	



(g) How long would it take one processor to complete all the tasks? 45

(h) Explain why the schedule you constructed in (f) must be an optimal schedule for two processors.

If we split tasks evenly between 2 processors it would take $45/2 = 22.5$ hours. So with integers, 23 is as good as we can do.

3. For each of the following, circle the correct answer and say a few words.

True False (a) The critical path algorithm always produces an optimal schedule.

The example in #1 on this sheet shows
this may not happen

True False (b) The decreasing time algorithm always produces a less-optimal schedule than the critical path algorithm.

The example in #1 on this sheet is one where the
decreasing time algorithm beats the
critical time algorithm.

True False (c) You can always find an optimal schedule.

There is some best schedule for any given
project and number of processors.

True False (d) Adding more processors will give you a shorter schedule.

Once $\frac{\text{total time}}{\# \text{ processors}} < \text{critical time}$,

there is no way adding more processors will
help.