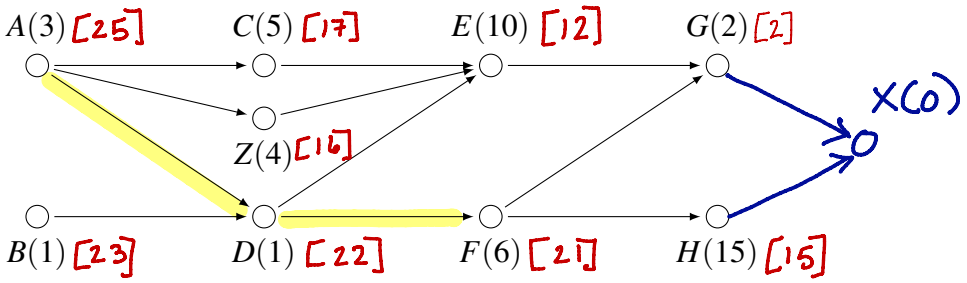
Goal Learn the Backflow Algorithm and, then, implement the Critical Path Algorithm.

1. **Backflow Algorithm**
   (a) Introduce an "end" vertex, say X, with a time of $[0]$.
   (b) From X, move back through every vertex assigning it the maximum time to reach vertex X.
2. Apply the Backflow Algorithm to the digraph below.



3. Use your work above to answer the questions below.
   (a) Find a critical path and the critical time in the digraph above. What do you observe about the numbers you produced in the Backflow Algorithm?

crit path: A D F H

time: 25

(The Backflow Algorithm finds this value, see vertex A!)

   (b) Determine the priority list using the Decreasing Time Algorithm.

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

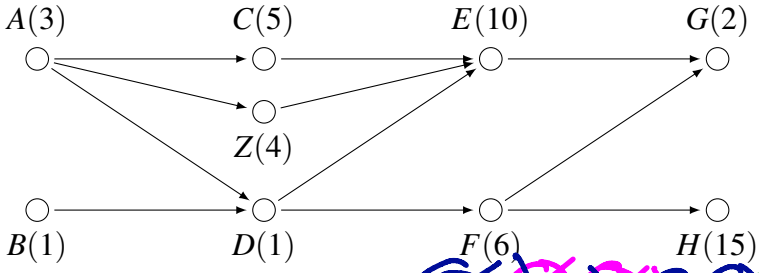
   (c) Determine the priority list by ordering the the vertices according to decreasing **critical time**.

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

4. In your own words, state the Critical Time Algorithm.

Create a schedule using a priority list obtained by decreasing critical times obtained from the Backflow Algorithm.

5. Use your two priority lists from the previous page to construct a schedule using two processors.



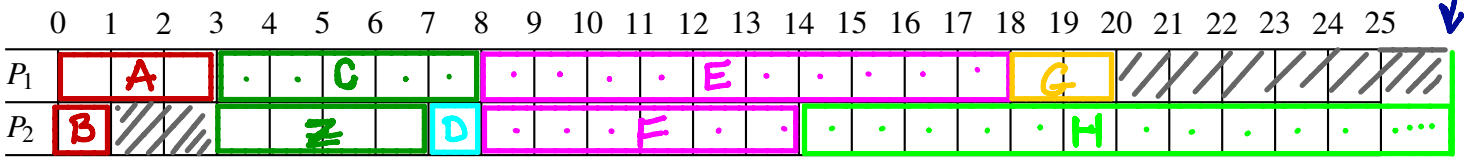
14
15
29

(a) decreasing time priority list: H, E, F, C, Z, A, G, B, D

time	0	1	3	7	8	14	18
ready	A, B		C D Z	D	E F	H	G
done		B	A	Z	C D	F	E

finishing time: 29
idle time: 11

29



(b) decreasing **critical** time priority list: A, B, D, E, C, Z, H, F, G

time	0	1	3	4	8	10	12	22
done		B	A	D	C	F	Z	E
ready	A, B		C D Z	E, Z	Z	H	E	G

finishing time: 25 ← clearly optimal!
idle time: 3