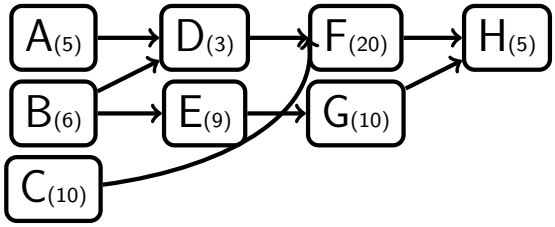
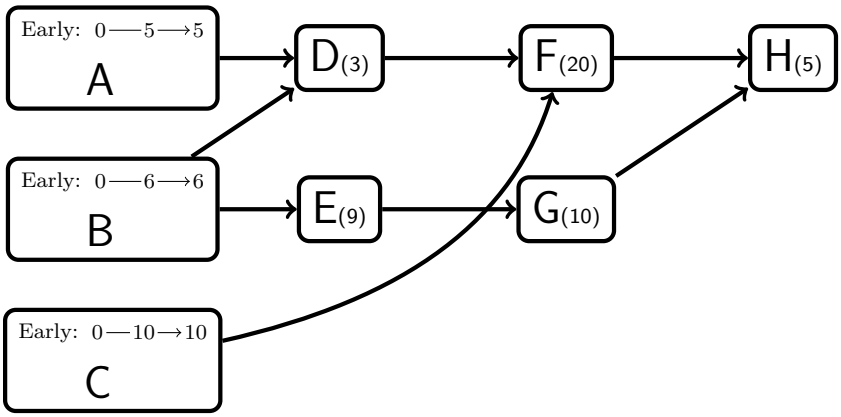
Here is another example with simpler names for tasks.

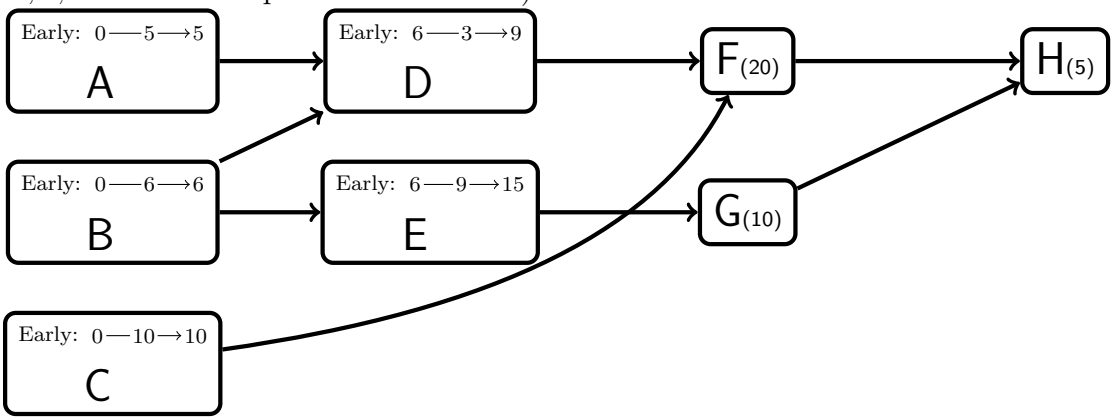
Task	Duration	Finish first
A	5	(none)
B	6	(none)
C	10	(none)
D	3	A, B
E	9	B
F	20	D, C
G	10	E
H	5	F, G



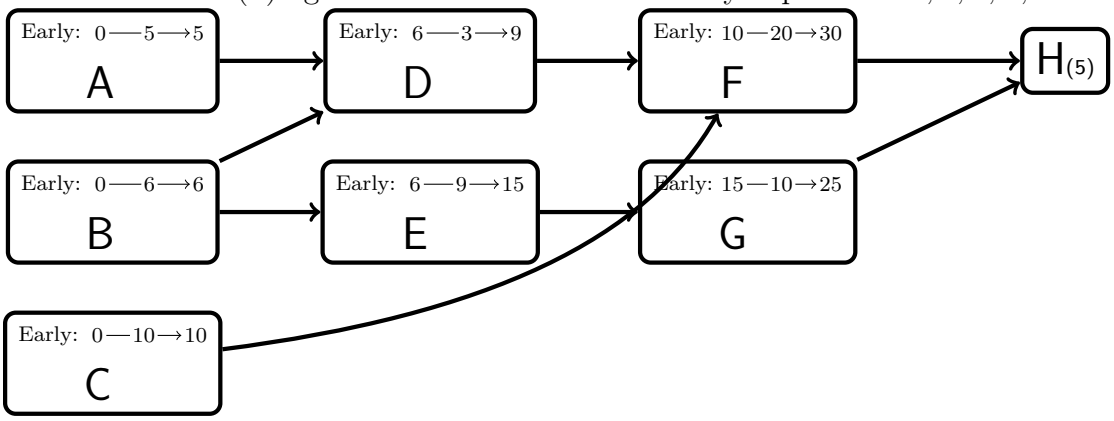
Then begin to fill in earliest start times, with all the "(none)" rows (all the bubbles with nothing pointing to them). This is rule (a).



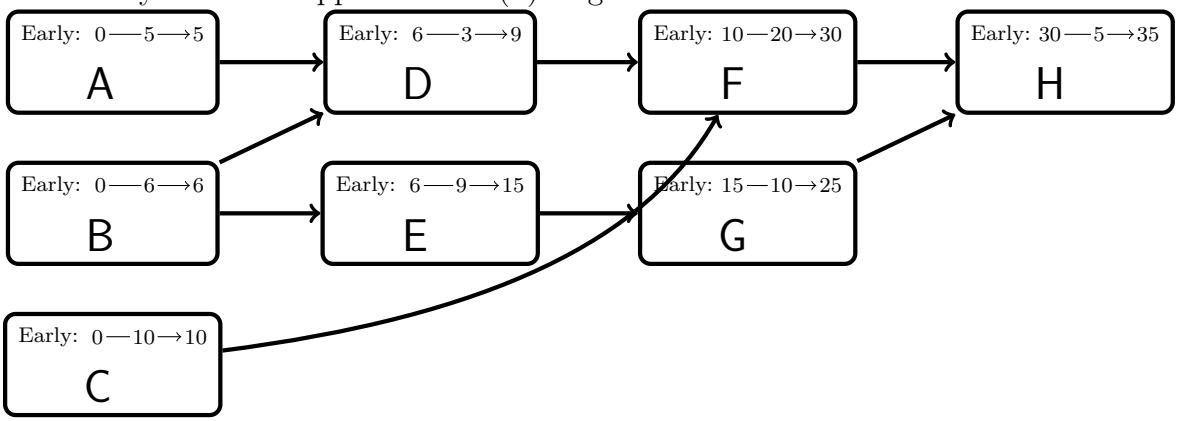
Now we use rule (b) to fill in the ones that only depend on A, B, and C's finish time (namely D,E; not F as it depends on D as well):



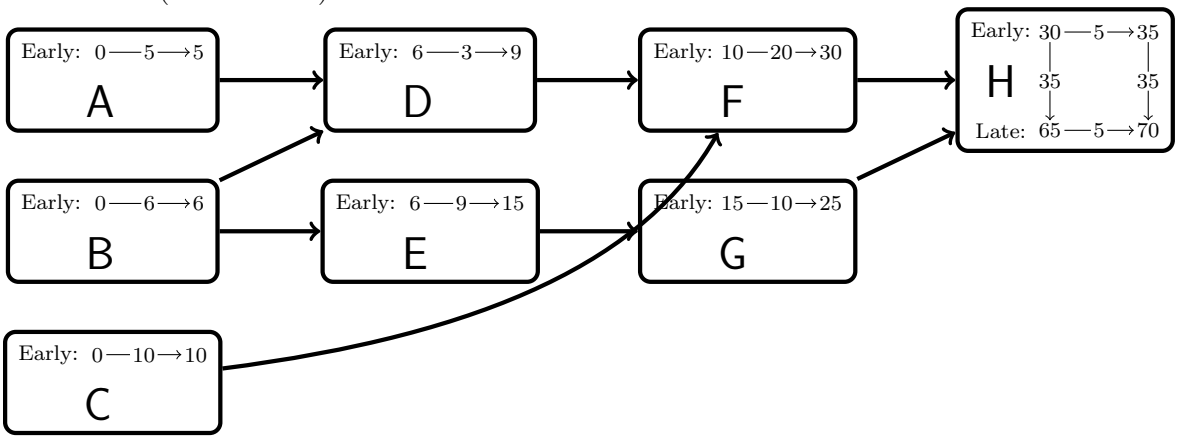
Now we use rule (b) again to fill in the ones that only depend on A,B,C,D,E:



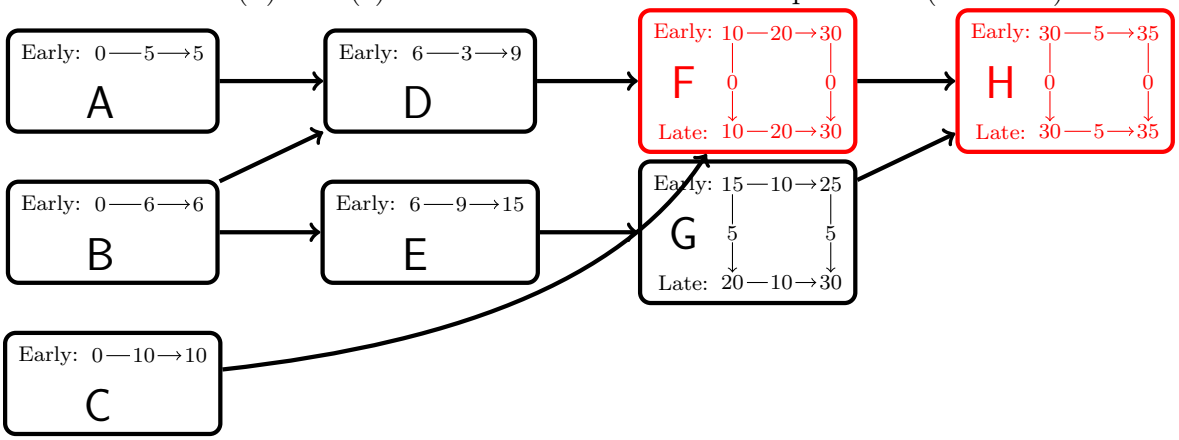
And finally one more application of (b) to get G:



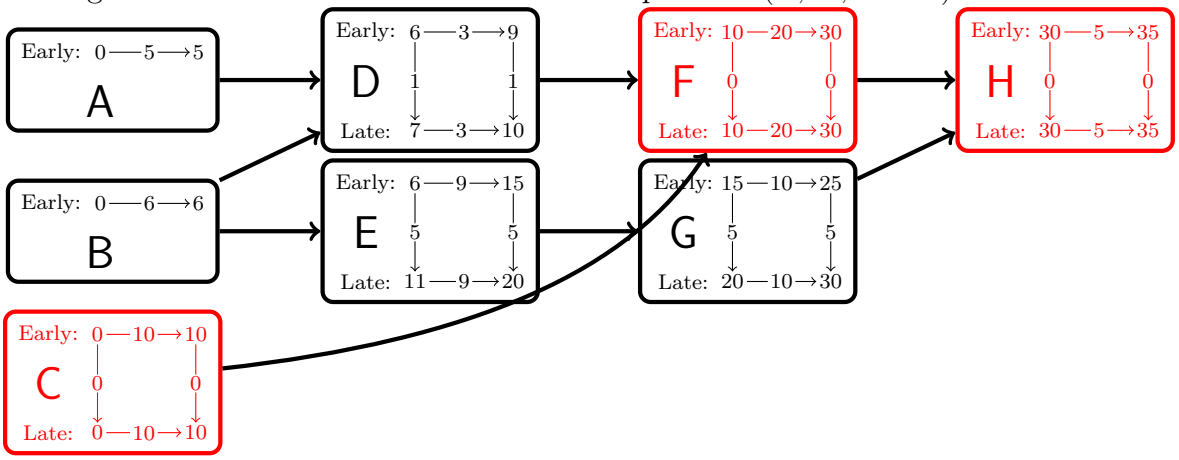
Now we switch to the latest start and finish times, based on the goal of finishing at 35 minutes (the maximum of all the earliest finish times). We use rule (a) to find H's latest finish time (35 minutes):



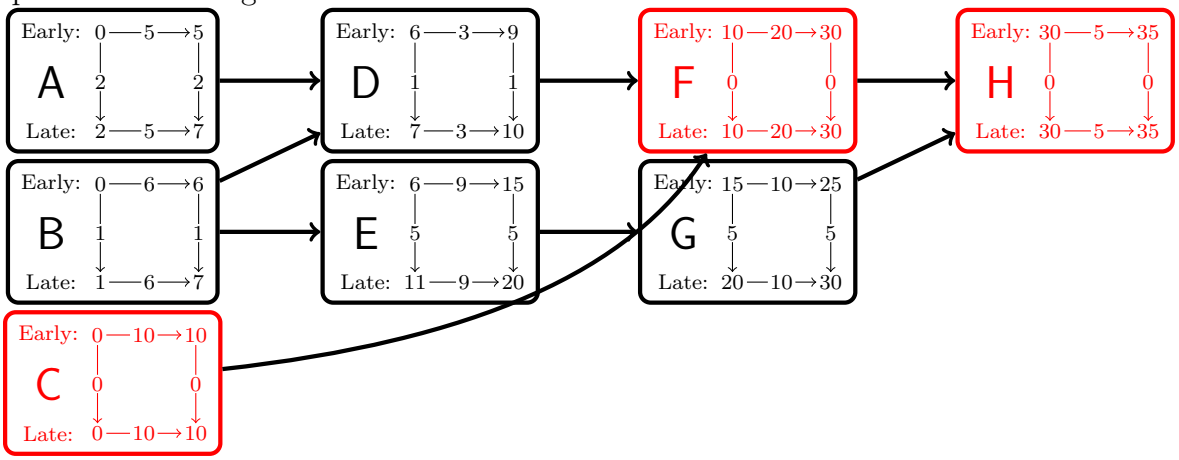
Now we use rule (b) and (c) to fill out the ones that H depends on (F and G):



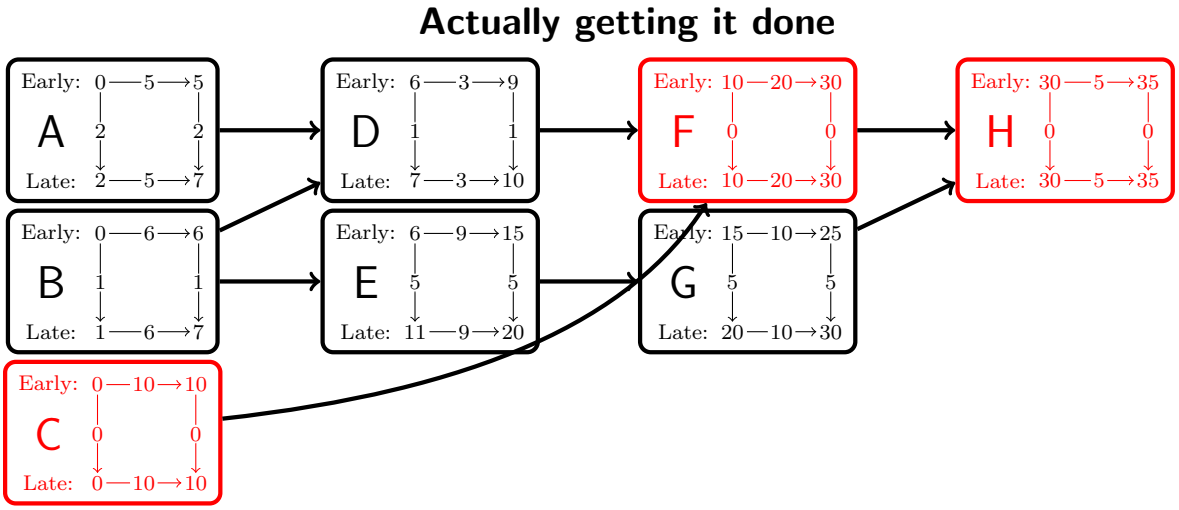
And again to fill out the ones that F and G depend on (C, D, and E):



The final one gives us A and B. Note how E is very flexible, but B still has to get done pretty quick in order to get D done on time.

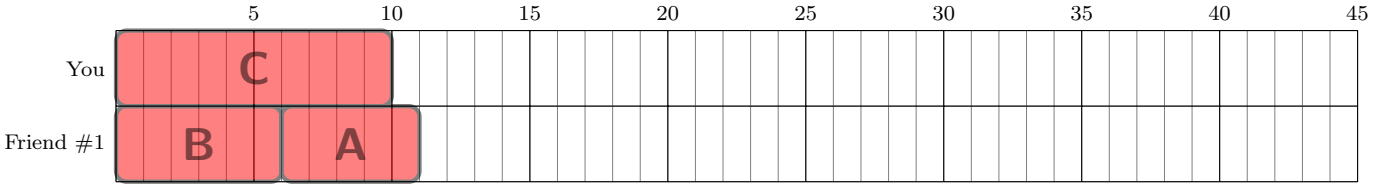


The critical path is $C \to F \to H$. Any delay in those tasks will delay the entire project. $B \to D$ is pressing as well. Any delay beyond a single minute in D (or B and thus D as well) will delay the entire project. Any delay in A beyond 2 minutes is bad, but $E \to G$ can wait 10 minutes with no issue. I would prioritize: $C > F > H > B > D > A > E > G$.

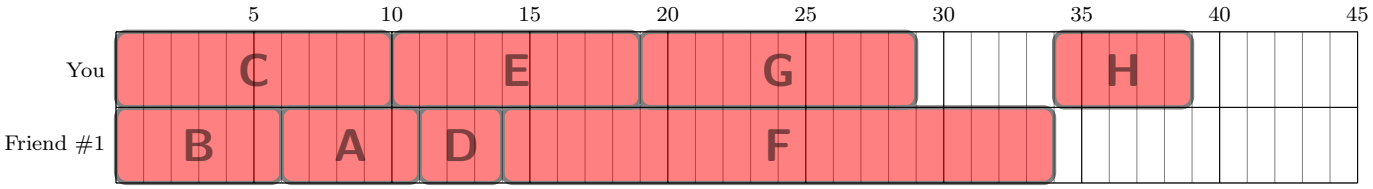


35 minutes suffices if we have unlimited resources. What if we can only handle two tasks at once? We use the priority list: $C > F > H > B > D > A > E > G$.

At first, only A, B, C can be done and C is most important. Your friend can work on something else, but the first thing in the priority list that can be done (other than C) is B. After 6 minutes pass, your friend is available, but the only tasks that can be done are A and E, with A higher priority. After 10 minutes we have:



Only E is available to be done at 10 minutes (since A is not yet done, D is not yet available) so we start E, even though it is low priority. At 11 minutes we can start D (and only D, since E is in progress). At 14 minutes D is finished, and we can start F. At 15 minutes, E is finished and we can start G. At 25 minutes, G is finished, but nothing else can start. At 34 minutes, F finishes, and now H can start.



What went wrong? Why did it take an extra 4 minutes? A was delayed by 6 minutes, but it only had 2 minutes of float time, a project delay of 4 minutes rippling along $A \to D \to F \to H$. Everything else was fine. B and C started immediately, and $E \to G$ started a minute ahead of "latest" schedule.