The key to a CPM problem is drawing a good network. Sometimes the key to drawing a good network is working in pencil, like in the case of this particular scenario, let's take a look. Consider the project information in the table below. This is a CPM problem because all it does is list activities, how long they are going to take, precedence relationship, and that's it. Actually, there's a little bit of instruction squeezed in here, draw and analyze the project network diagram, that's why I got hold of this pencil, to answer the following questions. Now, we'll do that, we will draw a network and we will analyze it, funny thing though, like if you were in a hurry, notice that the first questions here, it says, if you were to start this project which of the activities could you start on at the beginning the project. That one question, out of this scenario, you don't need to draw a network for, you don't need you any calculations, that one question is just about interpreting this data. If you were to start on the project, which activities could you start on right away? It's just asking you which activities do not have any immediate predecessors. That's just another way to ask that, so, without doing anything else, A B and C, at least I answered the first question in this scenario.

The second question in this scenario, how many paths are there through this project network? That's going to take a network diagram. Alright, just in case, it's kind of hard to tell how it will turn out, I'm going to switch to pencil. We always draw a network by drawing its start, ok, what do we do next? We look for, actually, what the question asked about, all the activities that don't have any immediate predecessors because they are here at the start. So I'm going to draw them, A B and C, got it, they're all here at the start. Now, basically, I start working my way down the list, ok. In terms of precedence information, who is mentioned next, okay, D has A as an immediate predecessor, I can see that, I can add that to the diagram because A is here, when you're done with A, I'm just drawing that fact, when you are done with A you can go on D, okay, then that one's done. Then I see that E F and G all have B as an immediate predecessor, so I might as well draw them, E, kind of like triplets, F and G all have B as an immediate predecessor. When you're done with B you can go on do any of these, ok, fine they are off the list. Let's see, H is next, H has C as an immediate predecessor, alright, C is down here, may as well add H, there's room. When you are done with C you can proceed to H, ok. Then I has G and H as immediate predecessors, task I, so I should locate it near G and H, this is task I, G H ok. And then there's only one other one, J has D and E, J D E okay, J will want to be near D and E, so to speak, so I'll put it here. J has D and E, ok at that point I have exhausted the list, I have all of the activities themselves, I have nodes or tables for each of the activities, the network's not done yet though, the network is not finished until, oh, the networks not finished until it has a finishing node. Just as every project has a start, every project has a finish, like a moment when the whole project is done, we need to draw that. The label, ok, finish, ok, we're done, oh, now this is important, who should be connected to the finish node? What you're looking for, you might want to review, is every activity that has arrows pointing at it but there's no way to leave it, like if you are traveling, see J, there's only arrows pointing at it, there's no arrows leaving it, that means it needs to be connected to the end. Now it's not the only one because if you look right here in the middle, F is the same way, it's a dead end right now, can't get out of F, that means it needs to be connected to the finish. And then down here you can get out, you can get out, I, there, the project network is drawn. Mow right here where I drew it, what actually was the question, it asked how many passes through the network, you're probably thinking, oh, we're going to have to do those early start, early finish, late start, late finish, times, yes actually, later, other questions, don't need to do that for this question. A path is any routes from start to finish, anyway, anyway to map from start to finish. To answer this question, why don't we just list them out when we find them. Paths, well let's see, if you are at the start, the first thing you can do is go up to A then to D then to J right, and from J that would get you to the finish, that's a path, ADJ, ok. Now what other options are there? Go back to the start, you could go to B, alright B and then, taking the high road, you can go to E, ok that will connect you to J, J, and then that will go to the finish, that's another path. Or go back to the start, you could go to B then go to F, from F actually that's a path, BF and then you would get to the finish, fine. Wait a minute, go back to B, there's another option, you go to G, G then to I, and I will connect you to the finish, that's a path. And then finally down across the bottom, could start out by going to C then to H, and then to I, that gives you the finish and then there's nothing else here. Alright, so, it's asking how many paths, I was listing what they were, but it just asks how many, that's one two three four five, there's five paths. So for the purpose of the question posed right there, that's the answer.

Oh but there's lots of other questions about this project following that. What is the critical path through the project? What is the completion time the project? Now we've got to work on the numerics. Alright, so, we're going to work on those questions right there. I'm going to go back, I need the network, I'm going to go back to the network and I'm going to start adding some detail to it. I'm going to start, kind of getting organized here, because these are really tables. So let me draw the dividers, they kind of look like flags, where you leave less room on the left hand side, more room on the right hand side, one of the things that that does, speaking of setup, is it leaves a neat little pocket, the lower left hand side, in which to copy the duration data associated with a task. A for instance, I'm looking it up, up here, is eleven days long, note that, it'll be handy later, A is eleven days long, B is ten days long, I'm just copying data onto the network, C is eight days long, D is thirteen days long, right, E is ten days long, F is ten days long, G is only two days long, that's good, H is six days long, I is seven days long and J is eight days long. Alright, the picture is all set up. Now, to analyze the network, and actually, to answer that second question that was posed, what's the completion time of the project, we need to find the earliest start, early finish times of each one of those tasks. How do you find early start and finish times? You go to the starting node, you go to the starting node and you note everybody who's connected to the starting node, ok, that's ABC in this case, good. Each one of them, their early start time is zero, because zero is right now, you have the option of starting on them right now, if you were a hurry, so in the case of A, zero plus eleven, if you are in a hurry, the earliest you could finish is in eleven days because you can start right now and it's eleven days long. B, zero plus ten, the earliest you can be finished is in ten days. C is zero plus eight, the earliest you could be finished is in eight days. Now what we want to do, is work our way across the network to the finish. Alright, go back up to the top, D cannot start until A finishes, that's what the network is telling us, A could finish at the earliest in eleven days, just pass that number over, that means D could start at the earliest in eleven days, it's waiting on A. Now eleven plus thirteen, it could finish, at the earliest in twenty four days, ok, right. How am I doing that? I'll do some scratch work over here, early finish time of a task is it's early start time plus the actual task time. Alright, now, Look at this center here, you have all these tasks that are waiting on B to finish, B could finish at the earliest in ten days, that means that ten is the early start time for each one of them E F and G, they could start at the earliest in ten days, right when B finished. The case of E, ten plus ten, the earliest it could finish is in twenty, the case of F, ten plus ten, the earliest it could finish is in twenty, in the case of

G, ten plus two, it could be finished at the earliest in twelve days from now. Now I'm looking at this one at the bottom, H, let’s see, H can’t start until one arrow trace it back, C finishes, C could finish at the earliest in eight days, then pass that number forward, that is the early start time for H, eight plus six is fourteen, H can finish at the earliest in fourteen days from now. We only have two other ones to do, lets look at J first, ok. J cannot start until, trace the arrows back, D and E finish, they are its immediate predecessors, now what do we know? D could finish, D in twenty four days at the earliest, and C could finish at the earliest in twenty days, the question is, what is the absolute earliest, if you are in a hurry that you could plan to start on J? Twenty four days, if you have a choice, there’s more than one immediate predecessor, you always pick the larger, because basically, you're waiting on both of them to finish, you can’t start until they both finish, so when can you start? Whatever the latest one is, the largest number, ok. So J could start at the earliest in twenty four days, ok, fine, twenty four plus eight, twenty four plus eight is what, thirty two, J could finish at the earliest thirty two days from now. That that we just reasoned through, our practice, is known is the early start time rule, the early start time of an activity is equal to the largest of the early finish times of all activities that immediately precede it. And I is another opportunity to practice that because, I has two immediate predecessors, G could finish at the earliest in twelve days, H could finish at the earliest in fourteen days, you pick the larger, I could get started at the earliest in fourteen days, fourteen plus seven what’s that, twenty one. I can finish at the absolute earliest in twenty one days. Now we've completed the early start and early finish times and I said, you know that will answer that business about what is the completion time of the project. Well it will if you know where to look for the answer. Ok, how do you determine, at this point, what the completion time of the project is? Go to the finishing node, finish node, trace back and identify all of the tasks that are associated, attached to the finish node, look at their early finish times. J could finish here at the end of the projects, see if you're attached to the finish node these are some, one of the last things to be done, J could finish at the earliest in thirty two days, F could finish at the earliest in twenty days, I can finish, at the earliest in twenty one days, these three things will be finishing up last, the project is not finished until everything is finished, therefore, when is the earliest the project can finish? Again you pick the larger, the thirty two, completion time of the project, now we know what it is, J told us, so to speak, completion time is thirty two days. Meaning that if we are smart, we could get this whole thing done in thirty two days. Ah, now, that didn't answer the question of what's the critical path through the project. We can’t answer that question until we find the late start, late finish times as well, and we can't find a late start late finish times unless, until we have that fact, what's the completion time. We said it was thirty two days, take that number, thirty two, this will get us started on the late start, late finish times, because that automatically becomes the late finish time of each activity, each one, thirty two, thirty two, and thirty two, that automatically becomes the late finish time of each of the activities that is associated with the finish node. Oh, because, we basically, in discovering it, had made the deadline, we’re saying that we could finish in thirty two days, therefore we're going to finish in thirty two days. Therefore, each task that’s attached to the end has to finish at the latest in thirty two days. Now what we want to do, was find these late start, late finish timings, which is to say, what's the latest you could start on something, and what's the latest that you can finish it and still stay on time. To finds them in general, we’re going to begin at the finish node, it's like we did here, and work our way back towards the start, ah because, just like in the case of J, it’s just logic, if we know it has to finish at the latest in thirty two days, thirty two minus eight, we know that has to start at the latest in twenty four days, right.

The late start time of an activity is late finish time minus the task time, think about it this way, we're now backing up for time. Here with F, thirty two minus ten, the late start time is twenty two. Here with I, thirty two minus seven, the late start time is what, twenty five, ok. Now, we wanna do this for all of them so we have to keep backing up. Go back to J, ok, we now know that J has to start at the absolute latest in twenty four days. It can't start until, I'm just reading the network, D and E finish, I'm tracing the arrows back, that means this is their new deadline. D and E, each one has to finish at the latest in twenty four days, ok. Now the subtraction, twenty four minus thirteen, D will have to start at the latest in eleven days. In the case of E, twenty four minus ten, it will have to start at the latest in fourteen days so as to not cause delays, ok. Now down here, I is sort of similar, we now know that it has to start at the latest in twenty five days, but it can start until G and H finish so we take that number and pass it back. Twenty five, twenty five, that's the latest they can finish, twenty five minus two, in case of G is twenty three, that's the latest it can start, H, twenty five minus six, the latest it can afford to start is nineteen days from now. Here we just keep working our way back, but as long as we're down here with H, right, it's connected then directly by working our way back to C, if it's got to start, at the latest, in nineteen days, you can just pass that number back, ok, that just means C gas to finish at the latest in nineteen days, nineteen minus eight, reveals that the late start time for C is eleven days. Notice that its early start time is zero, you have the option of starting right now, but what this is saying is that in the case of activity C you could also start as late as eleven days from now and it wouldn't delay the project, the project would still be thirty two days long, ok, good to know. Now we still have, up here's another fairly easy one because it's one to one, in the case of D we know that it has to start at the latest in eleven days, it is just waiting on A so you can pass that number back to the left, now that's the late finish time for A, it's gonna finish at the latest in eleven days, eleven minus eleven is zero. Oh, in contrast to C, activity A, the late start time is zero, the early start time is zero, the late start time is zero, I'm really relieved to see that because, here is a little tip, sometimes this arithmetic can get a little tricky, when you were working your way back, at least one of the activities that's attached to the starting node needs to zero out, the late start time needs to be zero, or result in zero, why do I mean it needs to be? Because if it doesn't, if you don't, if you have no activities at the starting node, that the late start time is zero, that's a signal that you messed up the arithmetic, ok, so, good. I see at least one, we're probably still on track. Now we still have one other task here to determine the late start, late finish, times, and this one we really should stop and think about this for a second, it's B, ok. EF and G cannot start until B finishes, that's what this means. What we now know, is that E has to start at the latest in fourteen days, F has to start the latest in twenty two days, and G has to start, at the latest, in twenty three days, therefore, in order to not delay any of these, trip any of these up, when does B have to finish at the very latest? Now you pick the smaller one, this is known as of late finish time rule. The late finish time of an activity is equal to the smallest of the late start times of all activities that come after it. You don't want to create any delays, so you have to pick the smallest this time, ok. So that's the fourteen, alright, so the late finish time for B is fourteen, and fourteen minus ten is four, the late start time for B is four days. So with B you could start right now if you wanted, its early start time or you can wait as late as four days from now, and that shouldn't be problematic theoretically, it wouldn't create any delays. Ah now, why did we do this? Well actually, publicly, for that one question, I said to find the critical path, it asked for the critical path, the critical path through a network is one of those paths that I listed up here, it's the set of all activities that have zero slack. Well slacks the difference between, like, the late start time, and the early start time, or the late finish time and the early finish time, so now you can see the critical path, because the critical path is the set of activities, that those two sets of timings when we worked through them, they matched, they match they are identical. Where do they match? Well it’s A they match, D they match, J they match, and then I should check, is this in fact, another check on my math, an actual path? Let me see, start, you can go to A, it does connect to D it does connect to J, and then that does connect to J, oh good, that is a path. Oh right, that’s actually earlier, we were just talking about paths, that was the first path that I mentioned, that's the critical path through this project. That's the critical path in this project, might as well officially note that, critical path is ADJ, that's the path that when you add up the task times along the path, its length is thirty two days, it is technically the reason the whole project will take thirty two days, well, anyway, it’s answered. What else is asked?

What is the earliest that activity G could possibly be completed? Oh that's easy, now that we've done all this other work, the earliest that G, that is the early finish time of G, it’s just asking us to go look that up. G, where’s G? G’s here, early finish, right here. Twelve, ok, twelve days. Ok, how about this one?

What's the most that activity F could be delayed in starting without delaying the project? Oh, that is also pretty easy given all the other work we've done because this is just another way of asking for delay in starting. Late start time and it’s asking about F, we just need to find it. F, F, F here, late, bottom row, start, right there, twenty two. In the case of activity F you could get started on that as late as twenty two days from now, and theoretically, if nothing else went wrong, you would still be finished with this entire project on time in thirty two days.