Edexcel A Level Further Maths: Decision Maths 1

Critical Path Analysis

Your notes

Contents

Activity Networks & Precedence Tables

Your notes

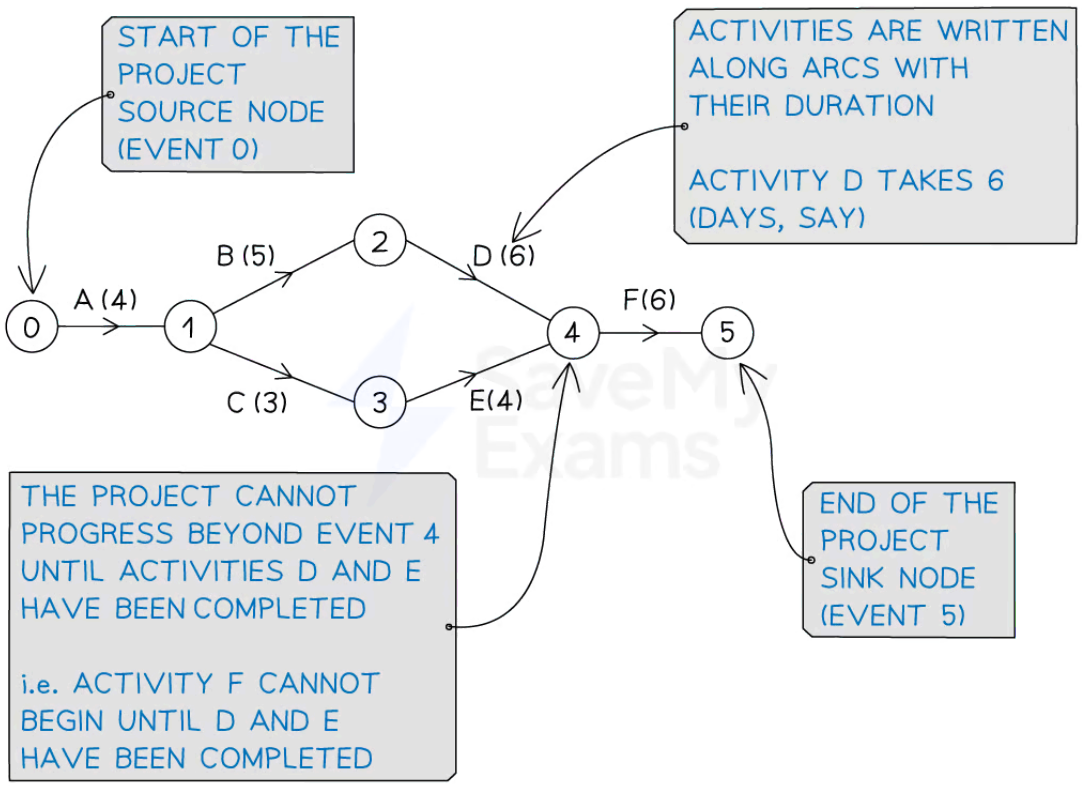
Activity Networks & Precedence Tables

What is an activity network?

- An **activity network** is a graph that shows the **activities** needed - and in what order - to complete a **project**
  - e.g. the project could be 'building a house' with activities such as 'foundations', 'walls' and 'roof'
- Some activities will depend on others being completed first
  - e.g. the activity 'foundations' would need to be completed before the 'walls' are built
- Some activities can occur at the same time
  - e.g. 'windows' and 'doors' can be fitted at the same time
- The **arcs** (edges) of the graph represent the **activities**
  - this may be referred to as an **activity-on-arc** network
- The **nodes** (vertices) of the graph represent **events** within the project
  - **events** can be thought of as **'stepping stones'**
  - the project cannot progress beyond an event until all the activities leading to that event are completed
    - 'you can't jump off the stepping stone until all activities leading to that stone are completed'

What does an activity network look like?

- **Events** (Nodes) are labelled with numbers, generally increasing in the direction of the project
  - The event at the **start** of the project is called the **source node**
    - it is labelled with **0** or **S**
  - The event at the **end** of the project is called the **sink node**
    - it will be the highest numbered node or labelled with **T**
- **Arcs** are labelled with their **activities,** with the **duration** given in brackets
  - activities are denoted by capital letters - ***A***, ***B***, ***C***, ***D***, etc
  - **arrows** are drawn on the **arcs** to show the order in which the project progresses
    - so strictly speaking, an **activity network** is a **directed graph**
    - in broad terms, this is generally from left to right across the activity network



What is a precedence table?

- A **precedence table** shows a list of the **activities** for a project
  - For each activity, the table includes a list of the **activities** that must already have been completed
    - but only the ***immediately* preceding activities** are listed, not all of them
  - Activities that do **not** have any precedents are indicated by '-'
    - these activities can begin at the start of the project
    - they will be attached to the **source node**

What does a precedence table look like?

- As well as a list of activities a **precedence table** may also show
  - the **duration** of each activity
  - the **number** of **workers** required to complete that activity

Your notes

Activity	Preceding activities	Duration (days)	No. of workers
A	–	4	2
B	A	5	1
C	A	3	2
D	B	6	3
E	C	4	1
F	D, E	6	2

ACTIVITY C CANNOT BEGIN UNTIL ACTIVITY A IS COMPLETE, ACTIVITY C TAKES 3 DAYS, REQUIRING 2 WORKERS

ACTIVITY E DEPENDS ON C BEING COMPLETE (THIS IMPLIES ACTIVITY A NEEDS TO BE COMPLETED TOO, BUT C IS THE IMMEDIATE PREDECESSOR)

Drawing an Activity Network

How do I draw an activity network?

- An activity network can be drawn from a **precedence table**
- Starting with the **source node**
  - add an **arc** for each activity, one at a time, considering its **immediately preceding activities**
  - an **event** (node) will be needed prior to each activity commencing
    - more than one activity can commence from the same event
    - more than one activity can finish at the same event
- A crucial feature of an activity network is that each activity has a **unique pair** of start and end nodes
- Any activities that do **not** precede another will go to the **sink node** at the **end** of the **project**
- In general, **activity networks**
  - use straight, arrowed lines for **arcs**
  - numbered circles for **events/nodes**

Examiner Tip

- A rough, curly-edged activity network often helps to start off with
  - This will give you a mental picture of what the network looks like
  - You can easily make changes, scribble bits out, etc with a rough diagram
  - When you are happy with it, you can redraw it neatly with straight edges
    - Remember the arrows, event/node numbers and activities with their durations!

Your notes

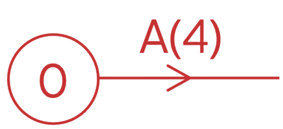
Your notes

Worked example

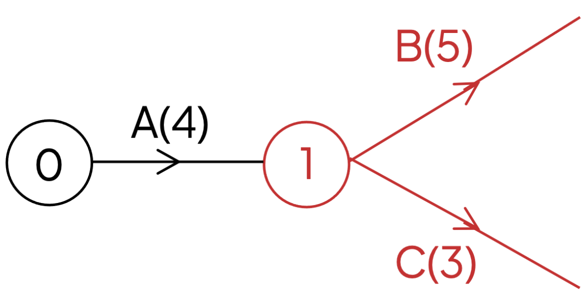
Draw an activity network for the precedence table given below.

Activity	Preceding activities	Duration
A	-	4
B	A	5
C	A	3
D	B	6
E	C	4
F	D, E	6

Starting with the source node, node 0, it is only activity A that can begin
So we will have one arc starting at the source node
Label the arc with an arrow, the activity name (A) and its duration (4)

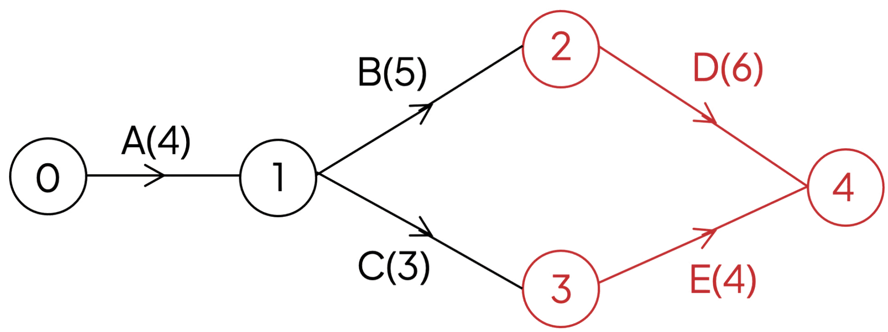


Activities B and C both depend on A, so add event/node 1 with arcs for B and C attached
Leave plenty of room (between B and C) in case anything later needs to go in between them

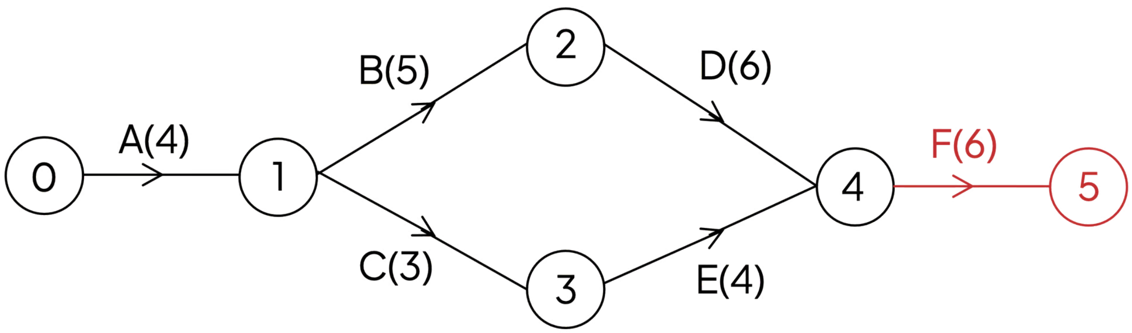


Activity D follows from B only, and activity E follows C only
Looking ahead though, activity F has D and E as immediate predecessors, so D and E need to meet at an event
So use event 2 to start activity D, event 3 to start activity E, and event 4 where they meet, ready for activity F

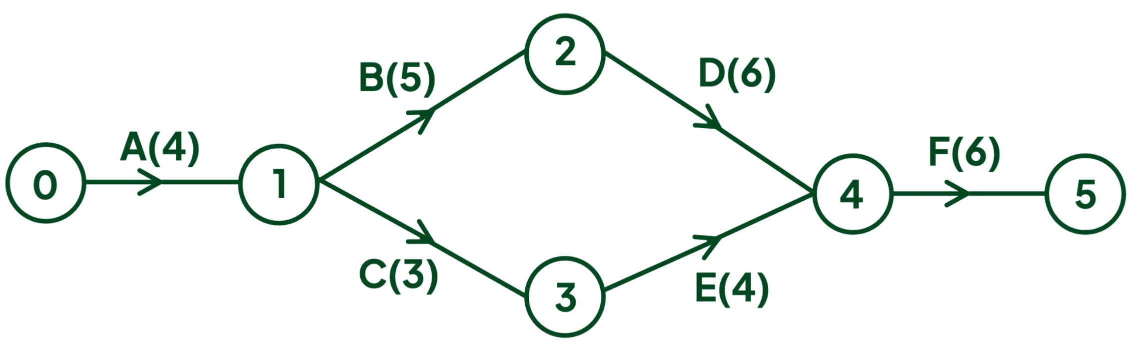
Your notes



Activity F is the last activity of the project so goes to the sink node, event 5



Check that all activities have a unique start and end node
For example, activity B starts at event/node 1 and ends at event/node 2 (this may be written as an ordered pair, (1, 2))
No other activity starts at 1 AND ends at 2 (C is (1, 3))
Checking everything else, the final answer is



Completing a Precedence Table

How do I complete a precedence table?

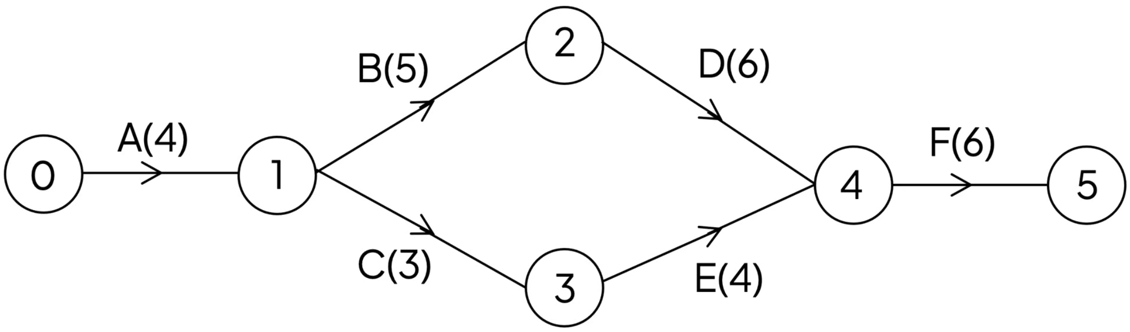
- A **precedence table** can be constructed from an **activity network**
- A basic table listing the activities and their duration can be constructed from the labels on the activity network
- To complete the **preceding activities** column in the table
  - start at the **source node**
    - any activities **starting** at the **source** node do **not** have preceding activities so use '-' in the table
  - for all other activities look at the event/node the activity **starts** at
    - any activities **ending** at this event/node are the **immediately preceding activities**
- The **numbers** of **workers** (for each activity) are not shown on an activity network so this column in the table would not be required

Your notes

Your notes

Worked example

Construct a precedence table for the activity network shown below.



The activities are A, B, C, D, E and F, with their durations given in brackets, so two columns of the precedence table can be completed immediately

Activity	Preceding activities	Duration
A		4
B		5
C		3
D		6
E		4
F		6

Starting at the source node 0, only activity A has no preceding activities, so this can be completed with a '-'
Work through each other activity considering the activities that go to its start event/node

- *activity B starts at event/node 1*
  - *activity A ends at event/node 1*
  - *B has immediate predecessor A*
- *C starts at 1*
  - *A ends at 1*
  - *C has immediate predecessor A*
- *D starts at 2*
  - *B ends at 2*
  - *D has immediate predecessor B*
- *E starts at 3*
  - *C ends at 3*
  - *E has immediate predecessor C*

- *F starts at 4*
  - *D and E end at 4*
  - *F has immediate predecessors D and E*

F is the last activity (it ends at the sink node, event 5) so the precedence table can be completed

Activity	Preceding activities	Duration
A	-	4
B	A	5
C	A	3
D	B	6
E	C	4
F	D, E	6

Your notes

Dummy Activities

Your notes

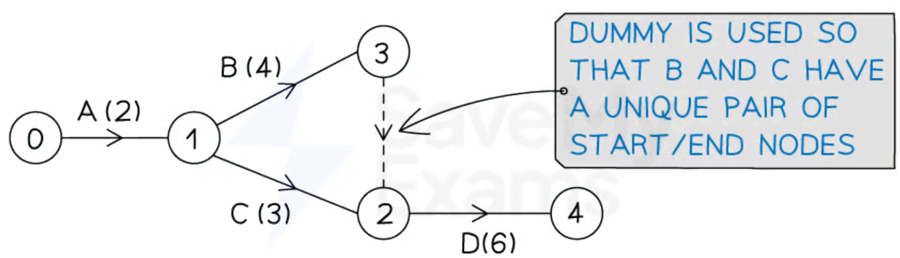
Dummies

What is a dummy activity?

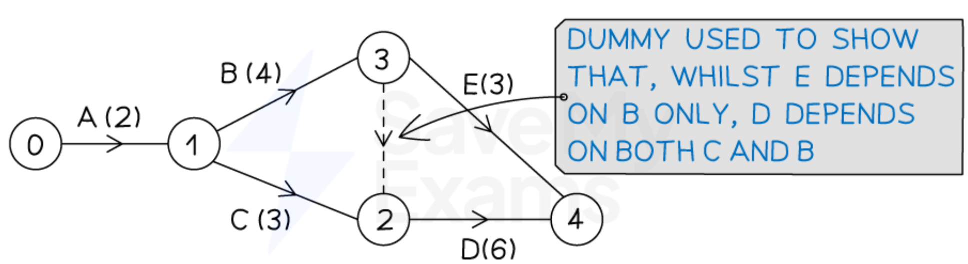
- A **dummy activity** is an activity that has a **weight** of **zero**
  - **dummies** are **not** assigned names (letters)
  - **dummies** are represented by **dotted lines**
- Dummies are used to show precedences in more complicated activity networks

When and where are dummies used in an activity network?

- **Two** situations can lead to the need for a **dummy activity**
- The **first** situation is to ensure each activity (arc) has a **unique pair** of **start** and **end nodes**
  - e.g. in the activity network below, activity D has **immediate predecessors** B and C
    - but B and C cannot **both** start at event/node 1 and end at event/node 2 (this would not be a unique pair)
    - a dummy activity is used so that B has start/end pair (1, 3) and C has start/end pair (1, 2)



  - Note that the dummy could also go from event 2 to event 3 with activity D commencing from event 3
- The **second** situation that requires a dummy is when there is a split of **immediate predecessors**
  - e.g. in the activity network below, activity D has immediate predecessors B and C
    - but activity E only has B as an immediate predecessor
    - a dummy activity is used to show that D depends on both B and C



Examiner Tip

- Exam questions will not always require you to draw the whole activity network
  - a diagram of part of the network may be given
- Exam questions are often specific about the number of dummies you should use
  - if you think you need more, go back to see if you can make improvements
  - it is generally expected that an activity network is as concise/efficient as possible with the minimum use of dummies

Your notes

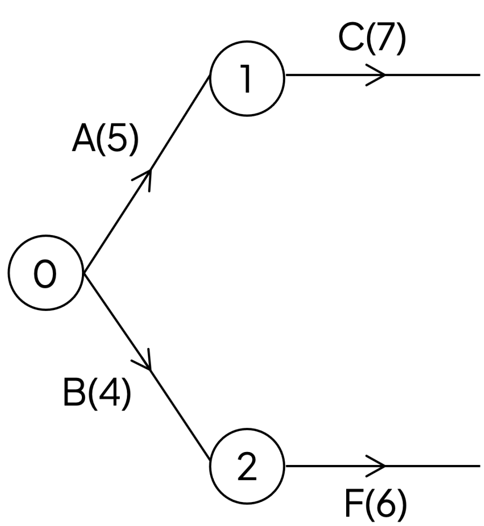
Your notes

Worked example

The activities involved in a project are listed in the precedence table below.

Activity	Immediately preceding activities	Duration (days)
A	-	5
B	-	4
C	A	7
D	B	3
E	A, D	7
F	B	6
G	C	6
H	C	4
I	G, H	5
J	E, F	4

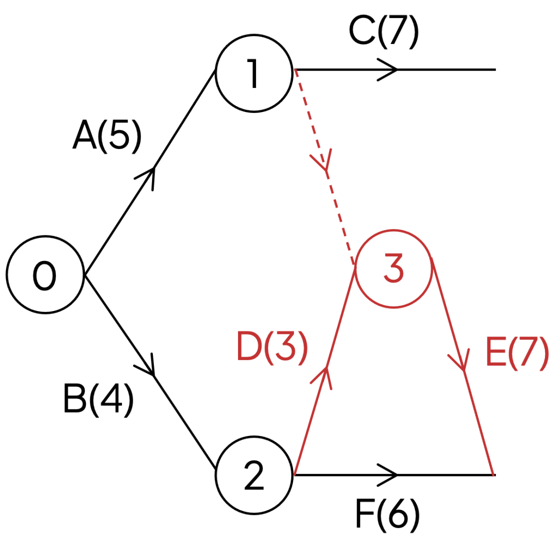
The project is also represented on the partially completed activity network below.



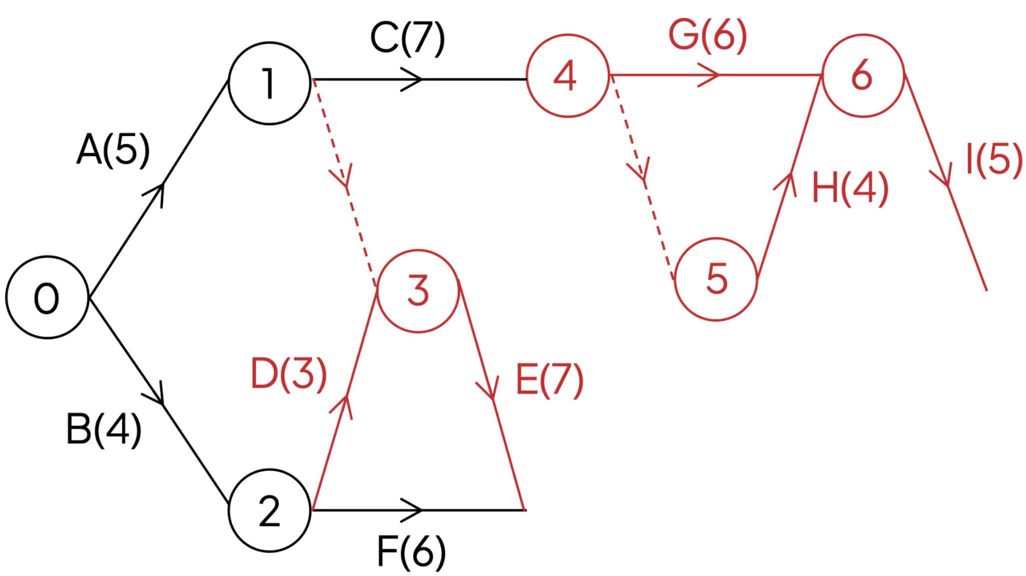
Using exactly two dummy activities, complete the activity network by adding activities D, E, G, H, I and J.

Activity D is dependent on activity B so draw an arc from event/node 2 for D
Looking ahead, activity E is dependent on both A **and** D, whereas activity C is dependent on just A

This is the second situation ('split predecessors') for the use of a dummy activity
Also from looking ahead activity J depends on both E and F - so the arcs for E and F will need to meet

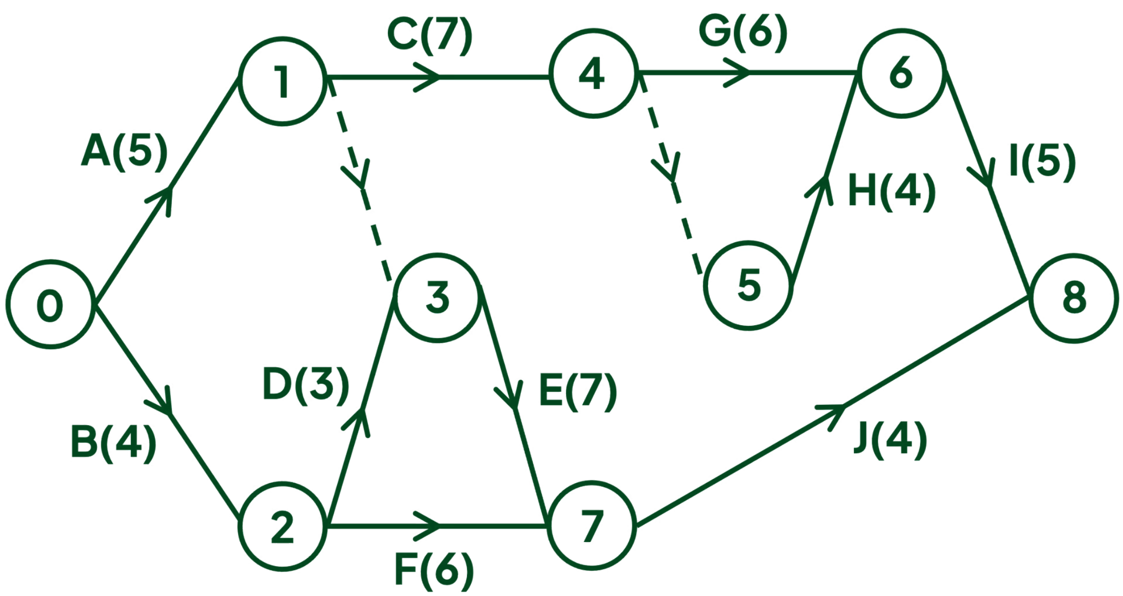


Activities G and H are both dependent on C, and activity I is dependent on both G and H
This could lead to G and H having the same start/end node pair
This is the first situation ('unique start/end node pair') for the use of a dummy activity
No activities depend on I so its arc can be drawn to the end of the project (sink node)



The activity network can now be completed with activity J and the sink node

Your notes



Your notes

Critical Path Analysis

Your notes

Introduction to Critical Path Analysis

What is critical path analysis?

- **Critical path analysis** (CPA) is the use of an **activity network** for a **project** that determines
  - the **minimum duration** of the entire project
  - the **critical activities**
    - these are the activities that **must** start and finish at the **earliest** possible time so that the project can be completed in its minimum duration
  - the **critical path**
    - the path from the **source** node to the **sink** node that encompasses all the **critical activities**
    - in harder problems there may be more than one critical path
  - any possible delay to starting non-critical activities
    - the **total float** of an activity is the time it can be delayed by as to not affect the minimum project duration

Your notes

Earliest Event Times

What are earliest (or early) event times?

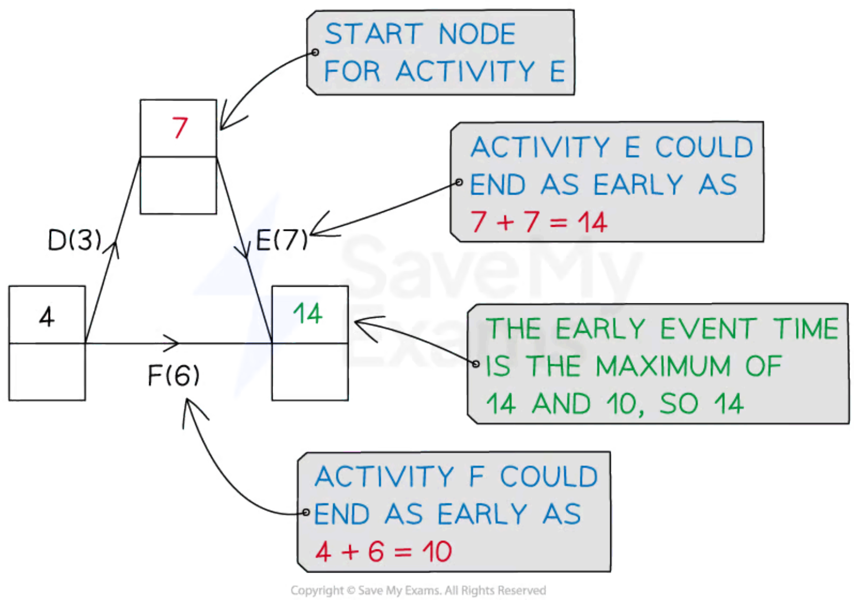
- The **earliest event time** is the **earliest** time at which an activity can commence from its **starting event**
  - this will determine the **minimum project duration**
- On an **activity network,** each **event/node** will have **two boxes** that require completing
  - sometimes the boxes will **replace** the circled event/node number
  - sometimes the boxes will be drawn **next** to the circled event/node number
  - the top box will be the **earliest event time**

  - the bottom box is for the latest event time - see the next section of this revision note
    - the **earliest event times** must be completed **before** the latest event times

How do I work out early event times?

- Starting at the **source node,** this will be 0
  - this is the starting point of the project, i.e. time zero
- ***Working forwards*** from the **source node** look at each of the **activities** leading ***to*** an event
  - (or, look at each of the **arcs ending** at a **node)**
  - for each activity's starting node, work out the **earliest time** it can **start**
    - this would be the **earliest event time** from its **start node add** its **duration**
  - the **earliest event time** is then the **maximum** of these times
    - this is the **earliest time** an **event** ('stepping stone') can be moved on from

$$\text{EARLY EVENT TIME} = \text{MAXIMUM}\left(\text{EARLY EVENT TIME AT ACTIVITY START NODE} + \text{ACTIVITY DURATION}\right)$$



- Completing the **earliest event times** is sometimes referred to as a **forwards pass** (through the activity network)

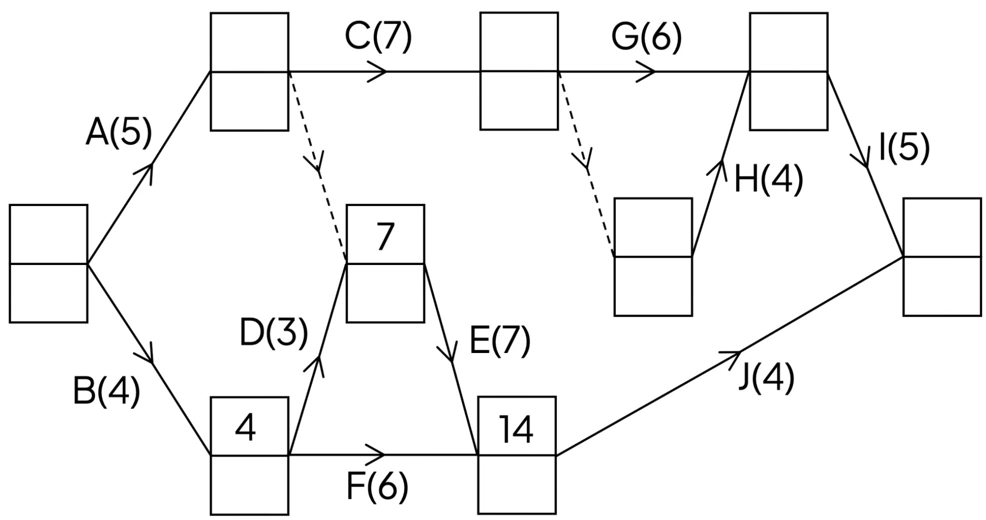
Examiner Tip

- An '**early** event time' is the '**latest** time' leading to a node !
- Ensure you check all arcs leading to a node
  - be particularly careful with dummies - these are unlabelled with a duration of 0
  - but still count as an arc leading to a node

Your notes

Worked example

Complete the early event times for the activity network given below.



Early event times go in the top box at each node
Start at the source node with early event time 0
Only activity A leads to the event (node) at the end of activity A

$$0 + 5 = 5$$

Similarly for activity C
(The earliest event time at the start of C is 5)

$$5 + 7 = 12$$

A dummy activity has a duration of 0, so the early event time at the end of the dummy will also be 12

Activities G and H lead to an event

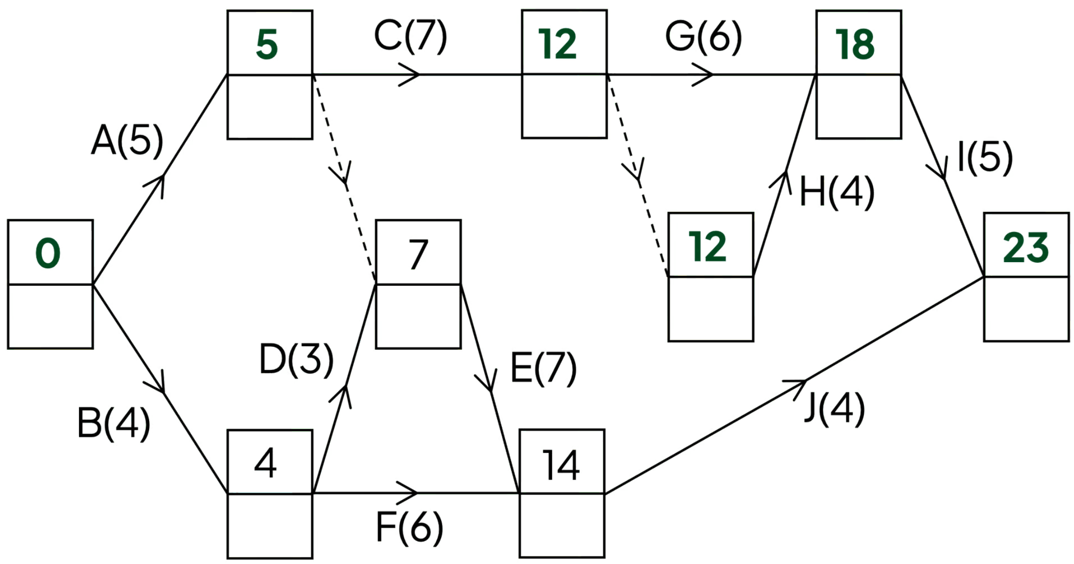
$$12 + 6 = 18$$
$$12 + 4 = 16$$
$$\text{Maximum } (18, 16) = 18$$

Finally the sink node

$$18 + 5 = 23$$
$$14 + 4 = 18$$

$$\text{Maximum }(23,\ 18) = 23$$

Complete the early event times on the network



Your notes

Latest Event Times

What are latest (or late) event times?

- The **latest event time** is the **latest time** at which an activity can commence from its **starting event**
  - without affecting the **minimum project duration**
- On an **activity network,** each **event/node** will have **two boxes** that require completing
  - sometimes the boxes will **replace** the circled event/node number
  - sometimes the boxes will be drawn near to the circled event/node number
  - the bottom box will be the **latest event time**

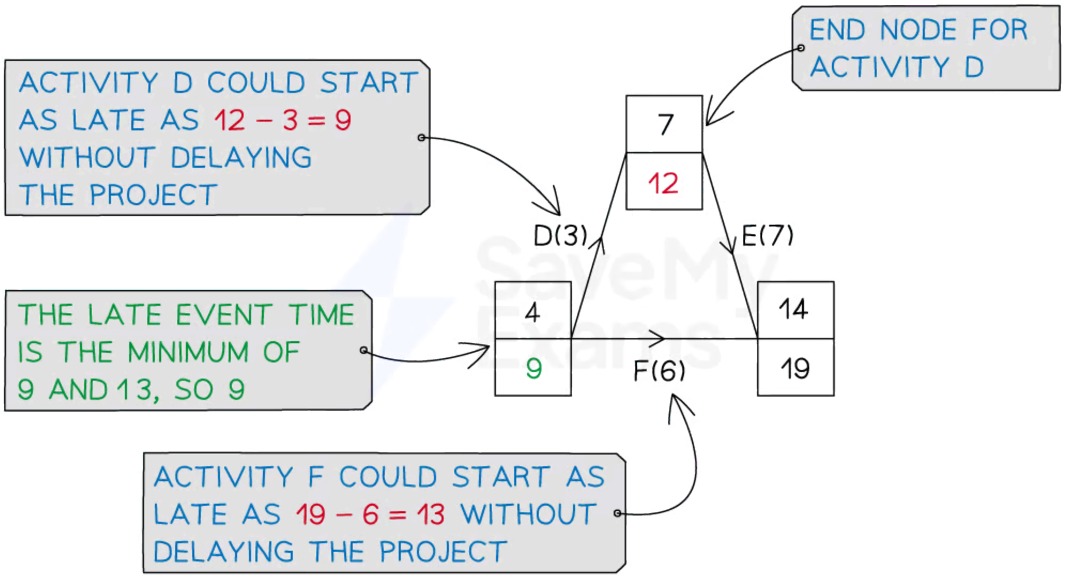
  - the top box is for the earliest event time - see the previous section of this revision note
    - the earliest event times must have been completed **before** the latest event times

How do I work out late event times?

- Starting at the **sink node,** this will be the same as the early event time
  - this is the **minimum duration** of the **project**
    - i.e. the shortest time in which **all activities** can be completed
- ***Working backwards*** from the **sink node,** look at each of the **activities** leading ***back*** to an event
  - (or, look at each of the **arcs starting** at a **node**)
  - for each activity's starting node, work out the **latest time** it can **start**
    - this would be the **latest event time** from its **end node subtract** its **duration**
  - the **latest event time** is then the **minimum** of these times
    - this is the **latest time** an **event** ('stepping stone') can be moved on from such that **all activities** can be completed in the **minimum duration** of the project

$$\text{LATE EVENT TIME} = \text{MINIMUM}\left(\text{LATE EVENT TIME AT ACTIVITY END NODE} - \text{ACTIVITY DURATION}\right)$$

Your notes

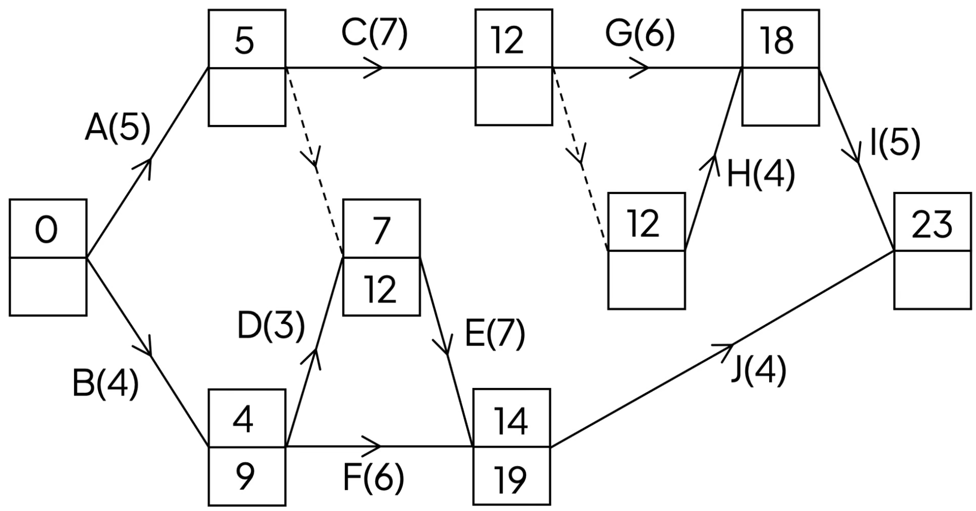


- Completing the **latest event times** is sometimes referred to as a **backwards pass** (through the activity network)

Your notes

Worked example

Complete the late event times for the activity network given below.



Late event times go in the bottom box at each node
Start at the sink node with late event time 23
Only activity I leads back to the event/node at the start of activity I

$$23 - 5 = 18$$

Similarly for activity H
(The latest event time at the end of H is 18)

$$18 - 4 = 14$$

Activities G and a dummy lead back to an event

$$18 - 6 = 12$$
$$14 - 0 = 14$$
$$\text{Minimum}\,(12,\ 14) = 12$$

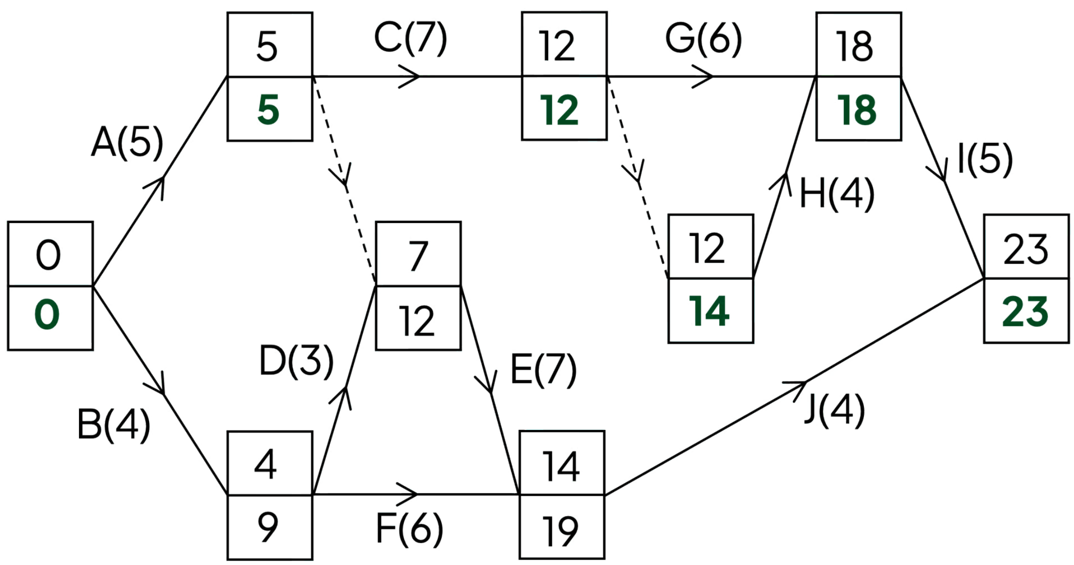
Activities C and a dummy lead back to an event

$$12 - 7 = 5$$
$$12 - 0 = 12$$
$$\text{Minimum}\,(5,\ 12) = 5$$

Finally the source node

$$5 - 5 = 0$$
$$9 - 4 = 5$$
$$\text{Minimum}\ (0,\ 5) = 0$$

Complete the late event times on the network



Your notes

Your notes

Total Float of an Activity

What is the total float of an activity?

- The **total float** of an activity is the time the start of an activity can be delayed by without **affecting** the minimum project duration
- **Critical activities** will have zero float

How do I work out the total float of an activity?

- For **non-critical** activities the total float can be calculated by considering the earliest time it can commence, and the latest time it can finish
- In words

TOTAL FLOAT OF AN ACTIVITY = LATE EVENT TIME AT ACTIVITY END NODE − EARLY EVENT TIME AT ACTIVITY START NODE − ACTIVITY DURATION

- Formally, you may see this written as

  The total float, $F(i, j)$, of activity (i, j), is defined to be $F(i, j) = l_j - e_i - \text{duration}(i, j)$,

  where e_i is the earliest time for event i and l_j is the latest time for event j

- Do not be put off by the word **'total'**
  - the **total float** of an activity is one value for one activity!

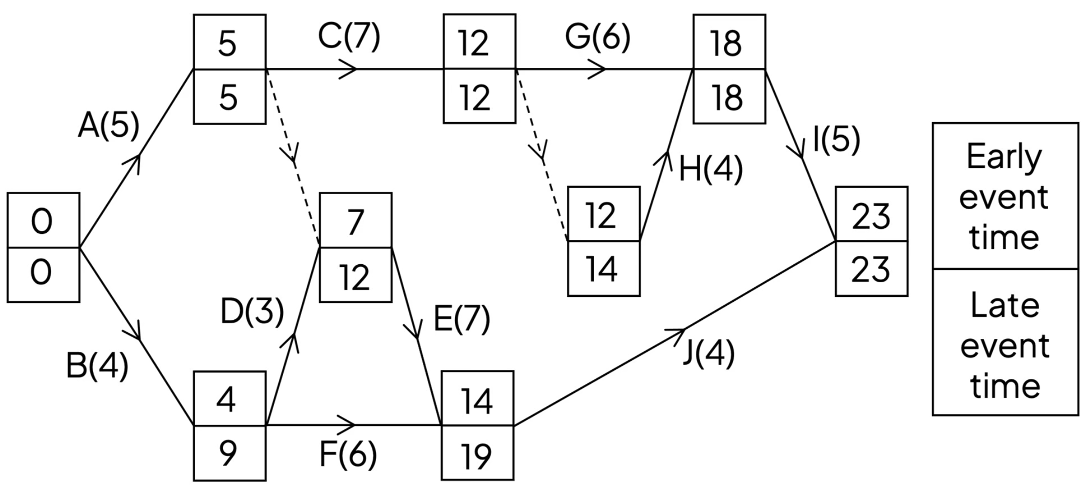
Examiner Tip

- In many cases, the total float can just be 'seen' or thought about by considering the activity network
  - but where things are complicated or busy, it is best to think about the formal way of calculating them

Your notes

Worked example

The activity network below shows the duration of activities A–J and the early and late times at each event.



Find the total float for activities B, D, E, F and J.

For activity B, the late event time at its end node is 9, the early event time at its start node is 0, its duration is 4

Total float of B = 9 - 0 - 4 = 5

Activity D has late event time 12 at its end node, early event time 4 at its start node and duration 3

Total float of D = 12 - 4 - 3 = 5

Similarly for activities E, F and J

Total float of E = 19 - 7 - 7 = 5

Total float of F = 19 - 4 - 6 = 9

Total float of J = 23 - 14 - 4 = 5

The total floats are B:5, D:5, E:5, F:9, J:5

Your notes

Critical Path & Critical Activities

What are the critical activities?

- **Critical activities**
  - have a **total float** of zero
  - must commence at their **earliest event time** in order for the **minimum project duration** to be achieved
- A **critical event** is an event where the **earliest start time is equal to the latest finish time**
- To find the **critical activities** look for the **critical events**
  - If an **activity is critical** then the start and finish events **must** be critical
  - However the converse is **not true**
    - It is possible for a n**on-critical activity** to be between t**wo critical events**
- Critical activities should be stated as a list
  - e.g. the critical activities are A, C, D and G

What is a critical path?

- A **critical path** through an activity network runs from the **source node** to the **sink node** such that the activities along the path have a **total float** of zero
  - Each activity in a critical path will be **critical**
- Where events/nodes are labelled with 0, 1, 2, etc the critical path would be given in terms of the events the path passes though
  - e.g. a critical path starting at source node 0, passing through events 2, 4, 5 and finishing at sink node 8 would be written 0 - 2 - 4 - 5 - 8
- Where events/nodes are not labelled, the critical path would be given in terms of the activities/arcs along the path
  - e.g. a critical path encompassing activities A, C, E, F and G would be written A - C - E - F - G
- It is possible that there is **more than one** critical path through an activity network
  - e.g. A-C-D-G and A-C-E-F-G could both be critical paths
    - In this case the critical activities would be A, C, D, E, F, G

How do I find the minimum project duration?

- The **minimum project duration** is the **quickest** time in which **all** activities in a project can be completed
- It is found by completing the **early event times** for each node in an activity network
  - the early event time for the sink node is the **minimum project duration**

Your notes

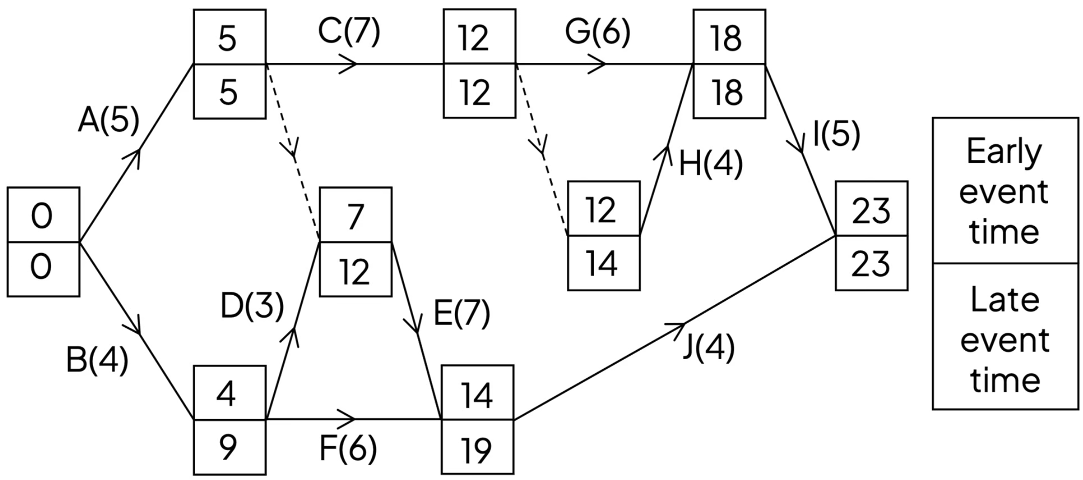
Examiner Tip

- Be clear about the difference between **critical path** and **critical activities**
  - these are practically found at the same time when analysing an activity network
  - make sure you write both as would be expected
    - the critical path would be a **sequence** of activities (sometimes written with '-' in between each)
      e.g. A - B - D - F - G
    - critical activities would be a **list,** with a comma between each activity
      e.g. A, B, D, F, G
- Some problems may state that all activities have the same duration (without specify how long)
  - The critical path and critical activities can then be found by looking for the path from source node to sink node that has the fewest edges
  - Alternatively you could just pick any number (>0) - 1 is the most obvious - for the duration of every activity and proceed as usual, but this would take far longer

Your notes

Worked example

The activity network below shows the duration (in days) of the activities needed to complete a project and the early and late times for each event.



Find the critical path, the critical activities and state the minimum project duration.

Start by looking at the nodes where the early and late event times are equal

Possible critical activities are A, C, G, I

Determine whether these activities have a total float of zero

Total float of A = 5 - 0 - 5 = 0
Total float of C = 12 - 5 - 7 = 0
Total float of G = 18 - 12 - 6 = 0
Total float of I = 23 - 18 - 5 = 0

Critical path is A-C-G-I

The critical activities are those that have a float of zero and lie along the critical path

Critical activities are A, C, G and I

The minimum project duration is the early (and late) event time at the sink node

Minimum project duration is 23 days

Gantt (Cascade) Charts

Your notes

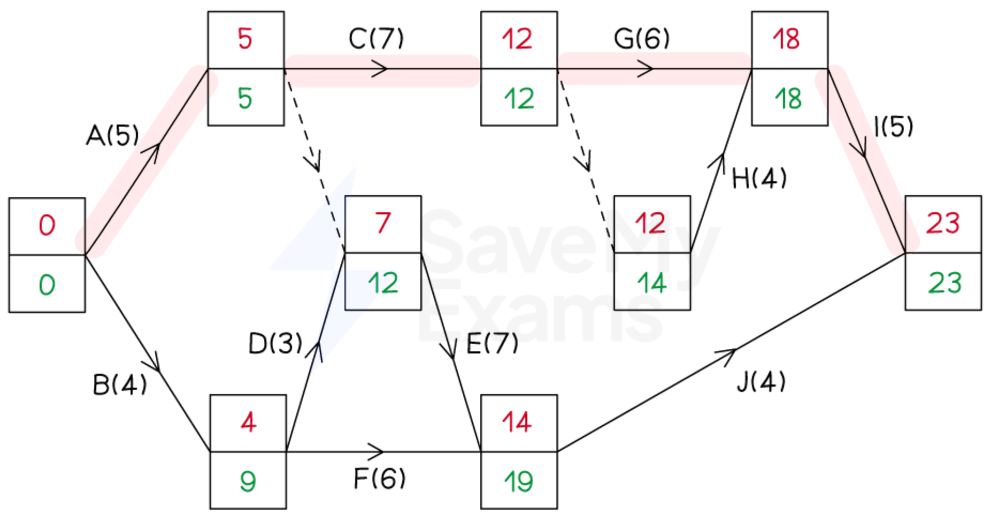
Gantt Charts

What is a Gantt (cascade) chart?

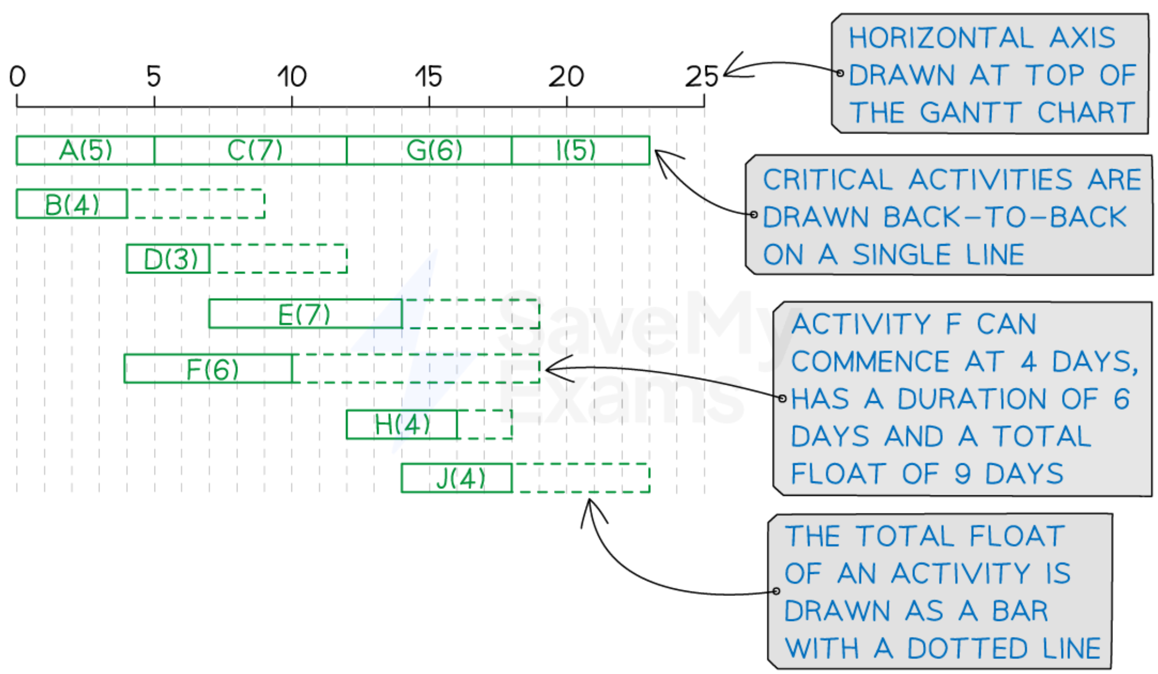
- A **Gantt chart** - also known as a **cascade chart** - is a graphical display of the **activities** making up a **project**
  - a Gantt chart shows
    - the **critical activities**
    - the **total float** for **non-critical** activities
    - the **minimum project duration**
- Gantt charts can be used in **resource levelling** and **scheduling** problems when the number of workers for each activity is known
  - in the first instance, a Gantt chart assumes **one worker per activity**

How do I draw a Gantt (cascade) chart?

- (Usually) a **horizontal axis** is drawn for time and this appears at the top of the **Gantt chart**
- **Activities** are then drawn as a series of **bars** (rectangles) underneath
  - each **activity** is assumed to **commence** at its **earliest event time**
    - i.e. at the early event time of its start node
  - each activity is assumed to occur in a single block of time
    - i.e. no breaks!
    - e.g. an activity of duration 5 and early event time 4 would be drawn as a bar starting at 4 and ending at 9
  - **Critical activities** are all drawn in the same horizontal line
    - these have a **total float** of **zero** so are drawn **back-to-back**
  - **Non-critical activities**
    - are drawn one per line
    - have their **total float** indicated by a bar drawn with a dotted line
    - e.g. an activity of duration 4, early event time 7 and total float 3 would be drawn as a (solid) bar starting at 4 and ending at 11, with a dotted bar starting at 11 and ending at 14
  - Bars are labelled with their activity name and duration
    - floats are not labelled
- For the activity network below
  - the **critical activities** are highlighted and are A, C, G and I
    - (the critical path is A-C-G-I)
  - the **minimum project duration** is 23 (days)



- The **Gantt chart** for the project would be constructed with
  - a **horizontal axis** running from 0 to (at least) 23
    - (0 to 25 keeps things nice!)
  - **critical activities** A, C, G and I drawn back-to-back on a single line underneath
  - activities B, D, E, F, H and J are each drawn on a **separate** line
    - e.g. activity D will be drawn as a solid bar from 4 to 7 with a dotted bar from 7 to 12



Examiner Tip

- An exam question is likely to provide a grid and the axes for you to draw a Gantt chart on
- For a non-critical activity the dotted float bar can be seen as room that the activity bar can slide back and forth along to vary it's start and end time

Your notes

Resource Histograms

Your notes

Constructing a Resource Histogram

What is a resource histogram?

- A **resource histogram** is a graphical way of showing the **number** of **resources** needed during each time unit (e.g. day) of a project
- A **'resource'** usually refers to a person (called a **worker)** but could refer to
  - a team of workers (e.g. bricklayers, roofers, painters)
  - a machine (e.g. cement mixer)
- The **number** of **workers** needed will vary as a project progresses and will be determined by
  - the **activities** that need to happen at a particular time
  - the **number** of **workers** needed per activity
- A resource histogram shows at a glance the maximum (and minimum) number of workers/resources needed for a project
- Resource histograms can be used to help minimise the number of workers at any particular time during a project

What assumptions are made in drawing a resource histogram?

- In the first instance, there are three main assumptions used in order to construct a resource histogram
  - Each activity begins at its **earliest event time**
  - A **worker/resource** can only carry out **one activity** at a time
  - Once an activity has begun, it **must** be **completed** in one go
    e.g. If an activity C, say, of duration 5 days, starts on day 7, it has to take place on days 7, 8, 9, 10, 11
    It **cannot** be done on days 7, 8, 12, 13, 14 for example
- Once a resource histogram is constructed, the assumptions can be reconsidered
  - This may be so that the number of workers/resources required at any given time is minimised
    - This process is called **resource levelling** and is covered in the next revision note

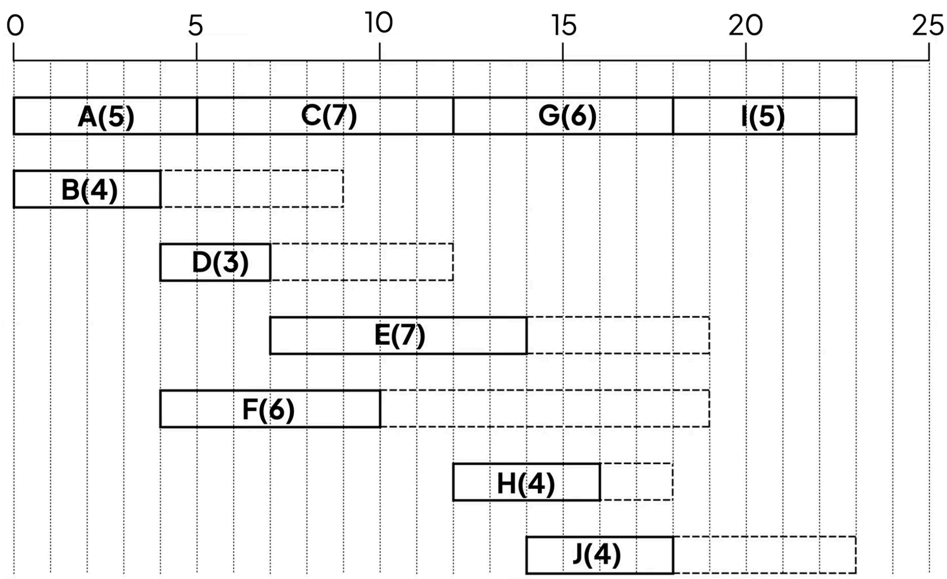
How do I draw a resource histogram?

- The **horizontal axis** is used for time
  - However units of time are treated as discrete data
  - So "day 1" would be labelled underneath the first column
- The **vertical axis** is used for the number of resources (number of workers)
- For each unit of time
  - a box per worker is drawn with a label of the activity that worker is doing
  - multiple boxes are 'stacked' on top of each other
    - critical activities are at the bottom of the stack
    - other activities appear in alphabetical order
- Each activity commences at its **earliest event time**

Your notes

Worked example

The Gantt chart for a project is shown below, with the duration of each activity measured in days.

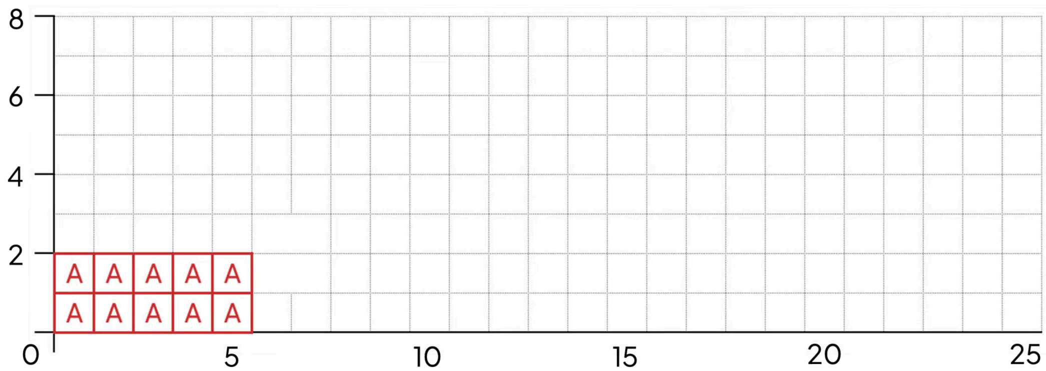


The table below shows the number of workers required for each activity.

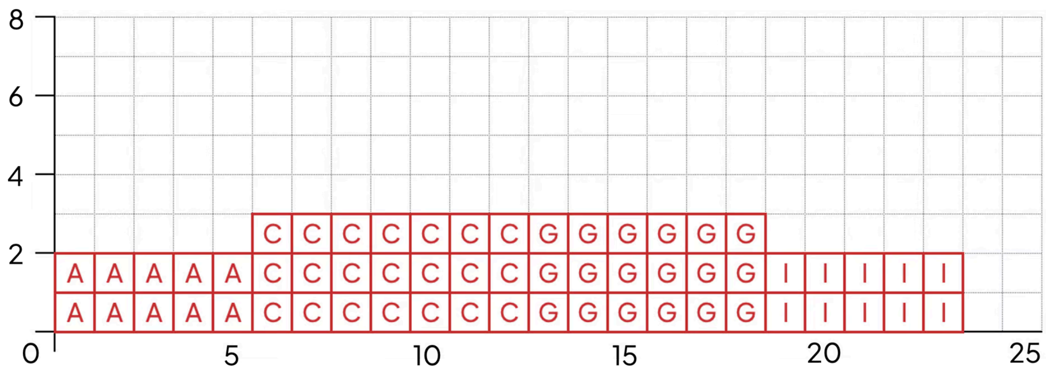
Activity	No. of workers
A	2
B	1
C	3
D	2
E	1
F	2
G	3
H	1
I	2
J	2

Construct a resource histogram for the project, assuming that each activity starts at the earliest possible time.

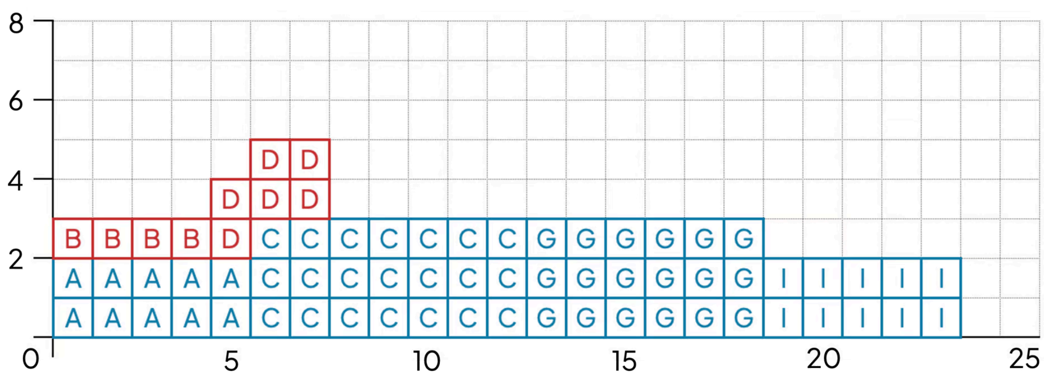
Activity A requires 2 workers and lasts for 5 days



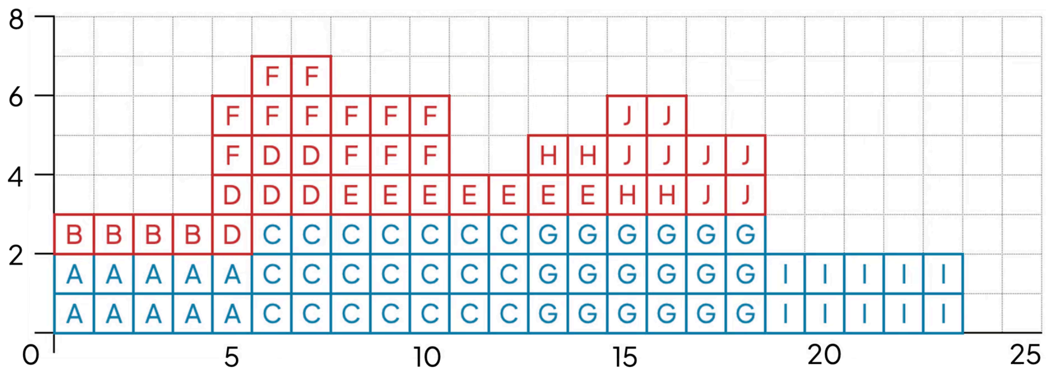
Add the rest of the critical activities with their number of workers



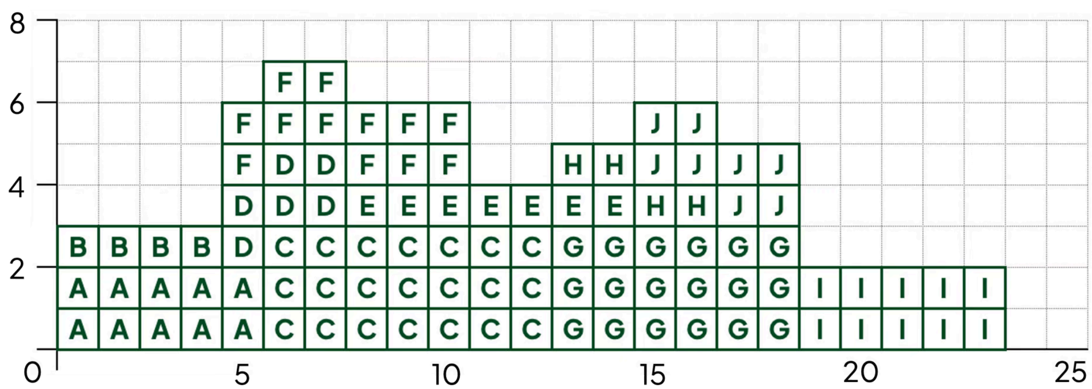
Other activities can be added in order
Activity B is straightforward but activity D gets 'bumped' up for days 6 and 7 days due to activity C requiring 3 workers (whilst activity A only needed 2 workers)



Add the other activities carefully, one at a time, considering where any are 'bumped' up or down



It is always worth double checking - especially any 'bumped' activities - and check the minimum duration time (23 days)



Your notes

Resource Levelling

Resource Levelling

What is resource levelling?

- Typically a project will have involve time periods when a **low number of workers** is required and other time periods when a **high number of workers** is required
- **Resource levelling** is the process of **analysing** a project, then **adjusting** the **start** (and end) times of activities with the aim of keeping the **number** of **workers** required at any particular time either to a minimum or within a specified **limit**
  - **Activities** from 'busier' times (lots of workers involved) are delayed to 'quieter' times (few workers involved)
    - i.e. workers (**resources**) are **spread** more evenly ('levelled') throughout the project

What does resource levelling involve?

- **Resource levelling** requires consideration of
  - **activity dependencies**
    - **precedence table** or an **activity network**
  - **early** and **late event times** and the **critical activities**
    - activity network and/or **Gantt chart**
  - the **number** of **workers** per activity
    - precedence table or a **resource histogram**
  - the **minimum project duration**
    - also called the **critical time** of the project
    - activity network or Gantt chart

What types of problem can be solved using resource levelling?

- The first type of problem requires **minimising** the **number** of **workers** needed at any particular time whilst **maintaining** the **critical time** of the project
  - Questions may ask which activities are occurring at certain times throughout the project
  - Phrases such as "on day 14" or "at time 8.5 hours" need to be interpreted carefully
    - This will depend on whether the resource histogram time axis is treated as **discrete** or **continuous**
- The second type of problem requires adhering to a **limit** on the **number** of **workers** but **minimising** the **delay** to the project's **critical time**
  - any delay to a **critical activity** would affect the project's **critical time**
- In both cases, an **optimal solution** is not necessarily required
  - but the solution needs to meet the **constraints** (restrictions/limits) regarding the **number** of **workers** and/or **project duration**

How do I apply resource levelling to a project where the critical time *cannot* be delayed?

Your notes

- **Start** by finding the time(s) with the **highest number of workers**
  - a **resource histogram** would be the easiest way of seeing this
  - remember that resource histograms - and Gantt charts - are initially drawn with activities starting at their **early start time**
    - i.e. the **early event time** at the activity **start node**
- Look for a **non-critical activity** (i.e. those activities that have a (total) **float)** such that
  - the **start time** can be **delayed** so that its **end time** will not surpass its **late end time**
    - i.e the **late event time** at the activity **end node**
  - This ensures there is no delay to the project's **critical time**
    - a (completed) **activity network** is the easiest way to find early and late event times
    - a **Gantt chart** is the easiest way to find a (total) **float** time
  - Delaying the **start** of an activity may have a 'knock-on' effect to any **other activities** that are **dependent** on it
    - check these activities can also have their **start times** delayed so that their **end times** do not surpass their **late end times**
    - again, this ensures no delay to the project's critical time
    - a **precedence table** or **activity network** are the easiest ways to determine activity dependencies

How do I apply resource levelling to a project where the critical time *can* be delayed?

- This is a similar process to above but as a delay is inevitable the **critical activities** *could* be delayed
  - The aim would be to **minimise** the **delay** to the project's **critical time**
- **Start** by finding the time(s) with the **highest number of workers**
- Consider the **non-critical activities** first
  - use the (total) **floats** as **efficiently** as possible so that **late end time(s)** are delayed by as little as possible
    - this means any **dependent** activities early start time(s) are delayed as little as possible
    - one-way to do this is to think about how activities can 'slide' back and forth on a Gantt chart
- If a **critical activity** *needs* to be delayed
  - consider any non-critical activities that are dependent on critical activities
    - they may become delayed too
  - Note that in problems where one or more of the critical activities are delayed
    - the **project minimum duration** (critical time) will change
    - and so (some of) the **critical path** and (some of) the **critical activities** will change

What is the lower bound for the (minimum) number of workers?

- In some cases, each activity will only require **one worker** (or one team of workers)
- If so the **lower bound** for the number of workers needed such that a project is completed in its **minimum project duration** can be calculated
  - It is found by finding the **smallest integer** that satisfies

$$\text{Lower bound} \geq \frac{\text{Total time of all activities}}{\text{Minimum project duration}}$$

Examiner Tip

- Exam questions may not use phrases such as **resource levelling**, **critical time**, etc
  - Practice problems to get a variety of experience of different contexts and phrasing

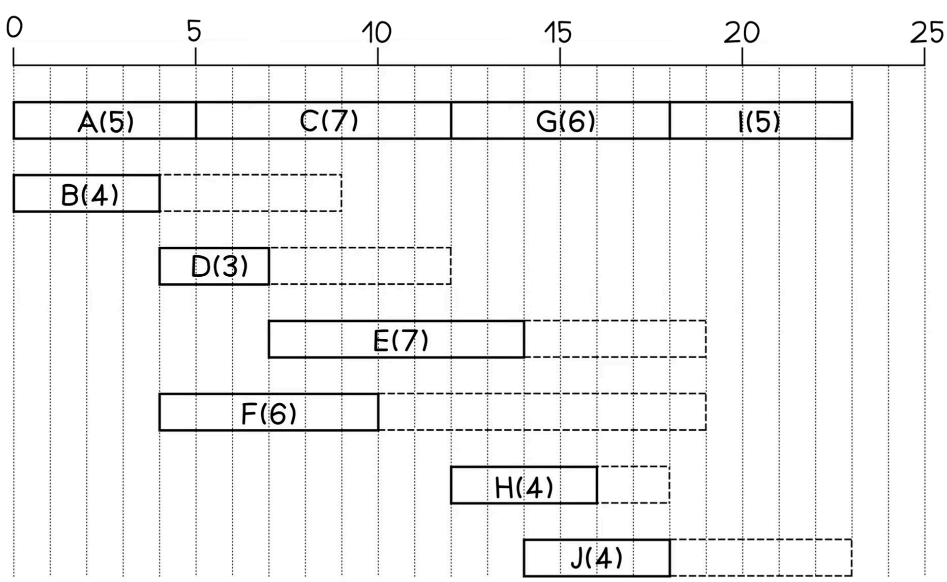
Your notes

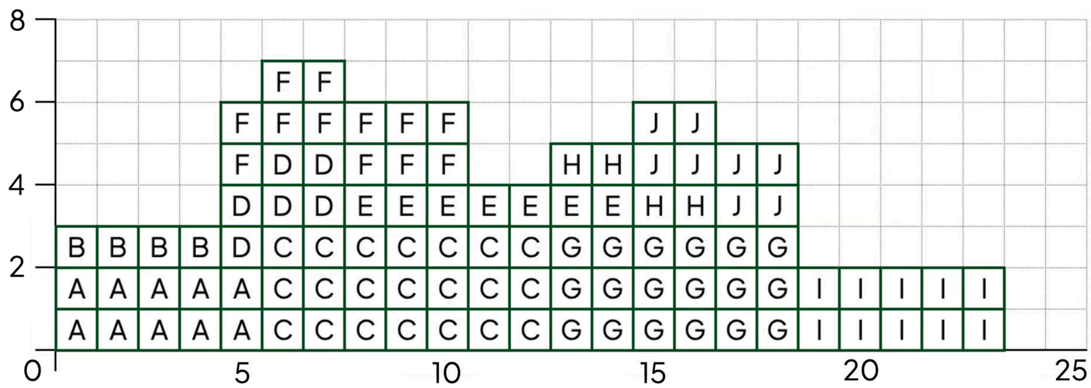
Your notes

Worked example

The precedence table, Gantt chart and resource histogram for a project are shown below.
Both the Gantt chart and the resource histogram show activities starting at their earliest event times.
Activity durations are given in days.

Activity	Immediately preceding activities
A	-
B	-
C	A
D	B
E	A, D
F	B
G	C
H	C
I	G, H
J	E, F





a) Use the diagrams to write down

i) the critical activities,
ii) the number of workers for each activity.

i) The critical activities are shown back-to-back in a single line on the Gantt chart

A, C, G, I

ii) The number of workers can be determined by the height of each activity on the resource histogram

A:2, B:1, C:3, D:2, E:1, F:2, G:3, H:1, I:2, J:2

b) Draw a new resource histogram to show that the project can be completed in 23 days without using more than six workers on any day.

The project is still to be completed in 23 days so the critical activities (A, C, G, I) cannot be delayed
The resource histogram shows that 7 workers are required on days 6 and 7 so consider the non-critical activities D and F on these days
Activity D has a total float of 5 days (late end time 12) and F has a total float of 9 days (late end time 19)

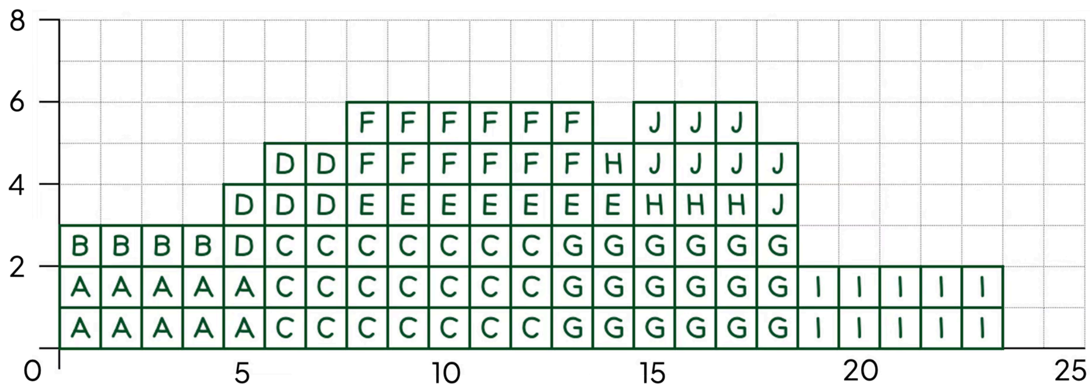
Delay F by three days

The resource histogram shows that delaying F by three days will mean that, without other changes, 7 workers would still be needed on day 13, but activity H has a (total) float of 2

Delay H by one day

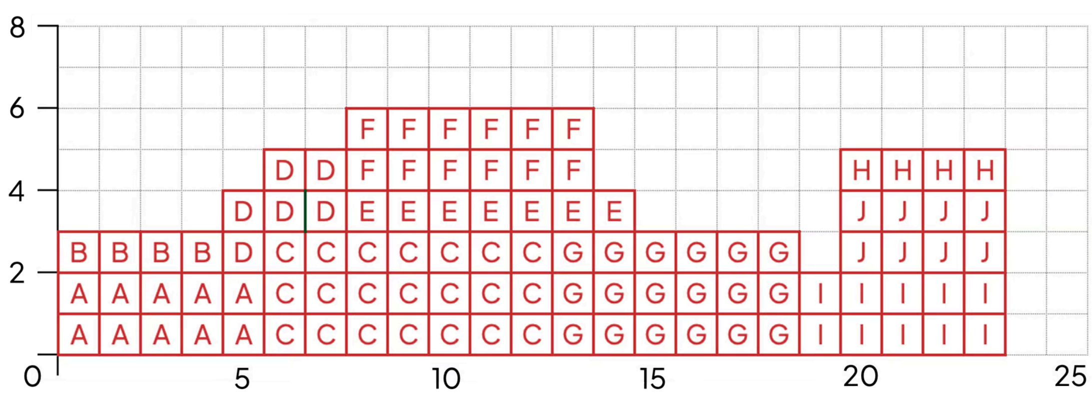
Delaying Activity H by one day - note how activity H does not depend on F but was still affected by delaying F - this is because we are concerned with the number of workers

The adjusted resource histogram is



c) Show, by means of a further resource histogram, that the project can be completed in 24 days with a maximum of five workers.

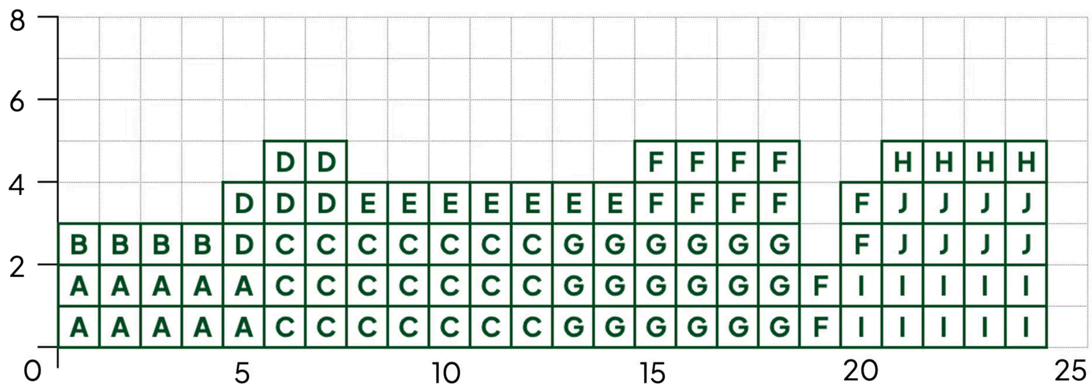
Using the solution to part b) (and the precedence table) we can see that activity J can be delayed and occur at the same time as activity I (with 4 workers)
Activity H only requires one worker so can also occur at the same time as I and J with 5 workers in total



To reduce the maximum number of workers from 6 to 5, activities C, E, F and G, E, F cannot happen simultaneously
Delaying activity F to start (immediately) after E will delay the project by one day
(6 days would be needed between E and J; J is dependent on F and there is currently only a 5 day gap)

The adjusted resource histogram, showing how five workers can complete the project in 24 days is

Your notes



Your notes

Scheduling Activities

Scheduling Activities

What is meant by scheduling (activities)?

- **Scheduling activities** is the process of **assigning workers** (resources) to **activities** within a **project**
- The **two** types of problem that arise involve
  - finding the **minimum number of workers** such that a project can be completed in its **minimum project duration**
    - The **minimum project duration** is also called the **critical time** of the project
  - finding the (new) **minimum project duration** given that their are **constraints** (restrictions) on the **maximum number** of **workers** available at any given time
- In harder problems, certain workers may only be capable of carrying out particular activities

What assumptions are made in scheduling?

- In **scheduling** activities the following assumptions are made
  - each **activity** requires only **one worker**
    - or one team of workers, i.e. one resource
  - an activity is assigned the first available worker
  - if there is a choice of activities to assign a worker to choose the activity with the **lower late end time**
    - i.e. the **lower late event time** at the **activity end node**
  - a worker can only work on one activity at a time
  - once a worker has started an activity, that activity needs to be completed

How do I schedule activities for a project that requires the *minimum* number of workers?

- For this type of problem, the **critical time** of the project *cannot* change
  - the **critical activities** cannot be delayed
  - **one worker** (resource) will complete all the **critical activities**
- Using a **Gantt chart** and considering the **non-critical activities**
  - **non-critical activities** can be delayed, but only **within** their (total) **float** times
    - i.e. **early start times** can be delayed but **late end times** cannot
  - **visualise** each activity as its bar being able to 'slide' (back and forth) within its box (solid and dotted)
    - aim for as few **overlaps** between activities as possible
  - activities can then be combined into as few rows as possible
    - by placing them 'back-to-back'
    - i.e. where possible activities should start **immediately** after others end
  - each **row** on the (reduced) Gantt chart will then be completed by **one worker**
    - i.e. the number of rows is the number of workers
- Remember that the **precedence** of **activities** needs to be maintained

Your notes

- e.g. Activity H, say, cannot move so it starts after activity I, as activity I depends on H being completed first
  - i.e. H is an **immediate predecessor** of I

What is the lower bound for the (minimum) number of workers?

- The lower bound for the number of workers needed such that a project is completed in its **minimum project duration** is the **smallest integer** that satisfies

$$\text{Lower bound} \geq \frac{\text{Total time of all activities}}{\text{Minimum project duration}}$$

- It is not always possible to schedule activities such that the lower bound can be met

How do I schedule activities for a project that has a *maximum* number of workers?

- For this type of problem, there will be a maximum number of workers available at any point in time
  - this *cannot* be exceeded, even if it requires activities to be delayed and the project's **critical time** increased
- Using a **Gantt chart**
  - find the **minimum number** of workers required to complete the project in its **critical time**
    - the Gantt chart would have already been largely reduced
    - do this using the process above
  - now consider how *any* activity (critical or non-critical) can be **delayed** such that
    - at any time the Gantt chart uses no more rows than the (maximum) number of workers available
- As in the first type of problem, the **precedence** of **activities** needs to be maintained
  - e.g. Activity H, say, cannot move so it starts after activity I, as activity I depends on H being completed first
    - i.e. H is an **immediate predecessor** of I

Examiner Tip

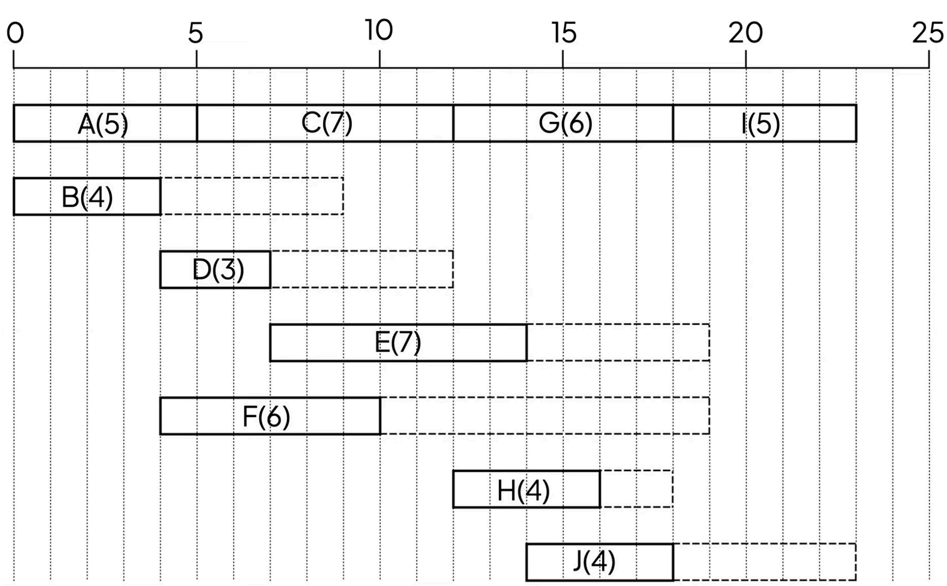
- Practice scheduling questions as exam preparation
  - there is not an algorithm as such and experience is the best way to become familiar with what to look for

Your notes

Worked example

A precedence table and Gantt chart for a project are shown below.
Each activity requires one worker and times are given in days.

Activity	Immediately preceding activities
A	-
B	-
C	A
D	B
E	A, D
F	B
G	C
H	C
I	G, H
J	E, F



a) Find the lower bound for the minimum number of workers required to complete the project within its critical time.

Your notes

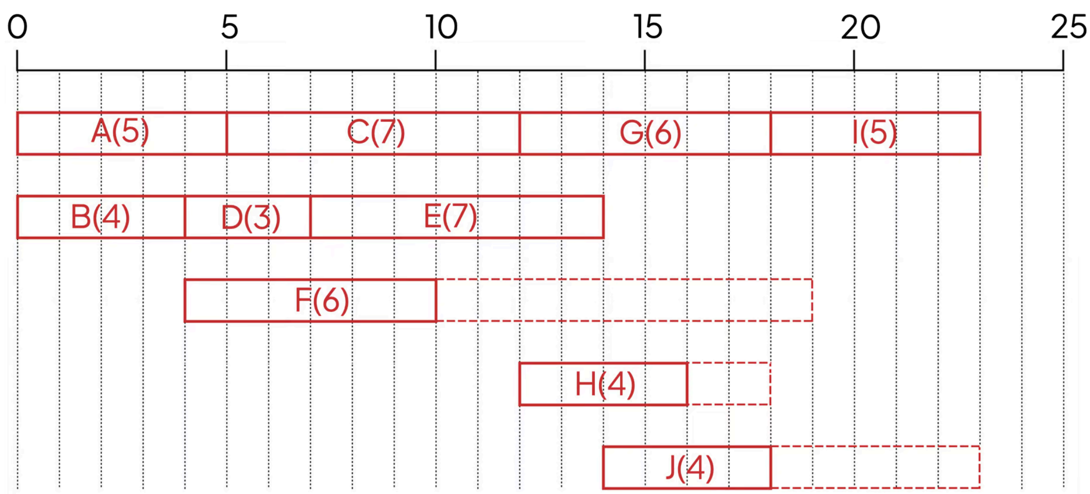
$$\frac{\text{Total time of all activities}}{\text{Minimum project duration}} = \frac{23+4+3+7+6+4+4}{23}$$
$$= \frac{51}{23}$$
$$= 2.217\ldots$$

The lower bound is the smallest integer greater than or equal to 2.217 ...

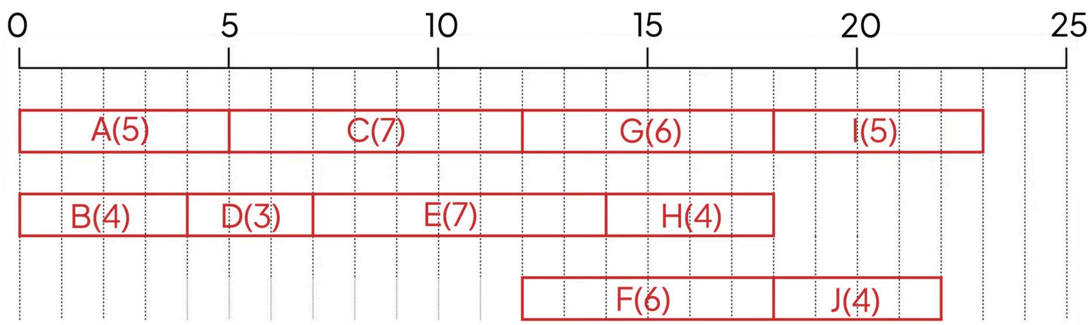
The lower bound for the number of workers is 3

b) Find the minimum number of workers required such that the project can be completed within its critical time.

Worker 1 ('row 1') will be assigned to all the critical activities ('back-to-back')
Worker 2 can start activity B at time zero, with activity D following immediately (at its early start time) and activity E immediately after that



By 'sliding' activities F and H it can be seen that there will be (at least) one day where either E and F or F and H will need to happen simultaneously - therefore a third worker is needed
There is some flexibility in assigning worker 3 but activity H is dependent on activity C
Activity J is dependent on E and F but as long as F finishes by its late end time that will look after itself
The adjusted Gantt chart is



The (minimum) number of workers is the number of rows in the adjusted Gantt chart, of which there are 3

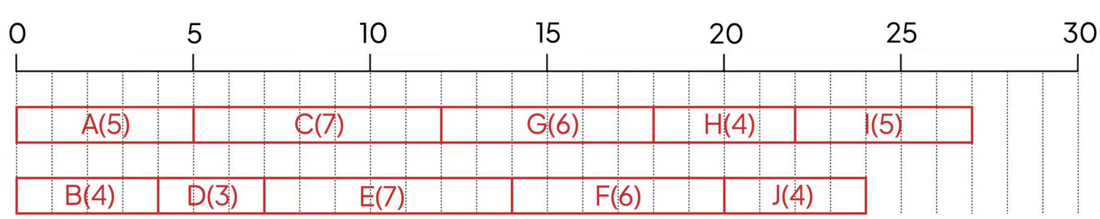
To complete the project within its critical time (23 days) a minimum of three workers will be required.

c) Find the minimum project duration given that a maximum of two workers are available at any time.

Using the solution to part b) we can see that activities A, C, B, D and E efficiently occupy two workers
Delaying activity F so it starts immediately after E will delay the whole project by one day (24 days)
(Be careful describing this, activity F would be starting ***after*** day 14, so starts ***on*** day 15)

This leaves activity H - I is dependent on it (and G) and H itself is dependent on C
Assigning H to worker 2 would delay the project by longer than necessary (at least 28 days) but by assigning H to worker 1 (either between C and G or between G and I) means precedences are maintained

The adjusted Gantt chart is



Activity I is the last to finish at a time of 27 days

The minimum project time for a maximum of two workers is 27 days.