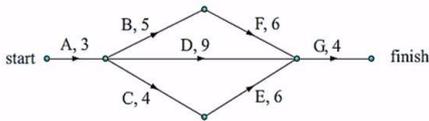




Goals Topic 3: Networks and decision mathematics



Activity Table

Activity	Duration	Predecessors	Earliest Start Time <i>EST</i>	Latest Start Time <i>LST</i>	Slack Time
A					
B					
C					
D					
E					
F					
G					

This brief:

- construct a network to represent the durations and interdependencies of activities that must be completed during the project; for example, preparing a meal (MAT18)
- use forward and backward scanning to determine the earliest starting time (EST) and latest starting times (LST) for each activity in the project (MAT19)
- use ESTs and LSTs to locate the critical path(s) for the project (MAT20)
- use the critical path to determine the minimum time for a project to be completed (MAT21)
- calculate float times for non-critical activities. (MAT22)

Theoretical Components

Resources:

For this week the theory work is in the *PDF file*:
Week 11-12 Notes & Exercises

Knowledge Checklist

- Shortest path
- Activity chart
- Network diagram
- Forward/backward scanning
- Earliest completion time
- Critical path
- Float time

Practical Components

There are questions to be answered in the booklet *Week 11-12 Notes & Exercises*.

Order of things to do:

1. Complete the questions in the booklet
2. Complete the investigation task
3. Submit the questions for marking
4. Complete the online Mathspace task

Investigation

On HawkerMaths and attached to this brief.

MATHEMATICAL APPLICATIONS 4

WEEK 11-12 NOTES & EXERCISES

Shortest paths

Given a network representing the distance between towns, consider the question, 'How far is it from town A to town X?' Such a question can be approached using a trial-and-error method. However, when networks become more complex, a systematic method is required. This method used is called the shortest path algorithm.

Shortest path algorithm

To find the shortest path between A and X in a network, follow these steps.

Step 1: For all nodes that are one step away from A, write the shortest distance from A inside the circle representing the closest node.

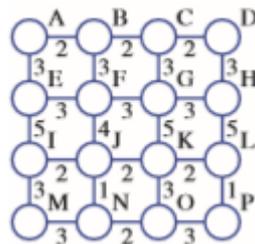
Step 2: For all nodes which are two steps away from A, write the shortest distance from A inside the circle representing the closest node two steps away.

Step 3: Continue in this way until X is reached.

Step 4: The shortest path can be identified by starting at X and moving back to the node from which the minimum value at X was obtained, then continuing this process until A is reached.

Example

Find the shortest path from A to P in the network. The units are in minutes and represent time taken. **Note:** We have placed the labels outside the nodes so that the times can be placed inside the circles.



Solution

Beginning at A, write down the shortest time taken to get to B and E inside each node.

Then write in the shortest time for all nodes which are two steps away from A. That is, C = 4, F = 5 and I = 8. Continue in this way until P is reached.

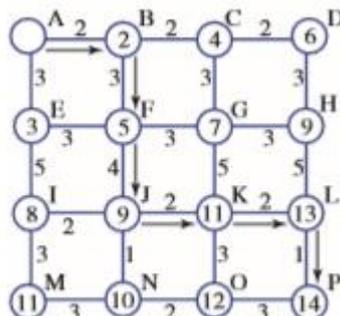
For example, at node J, the time from node I would be 10, so the shorter time, 9, from F is written in the node.

Now backtrack from P moving from node to node along the arcs which produced the minimum values.

Check to see if this is the shortest path. If this is the shortest path, put arrows on this path.

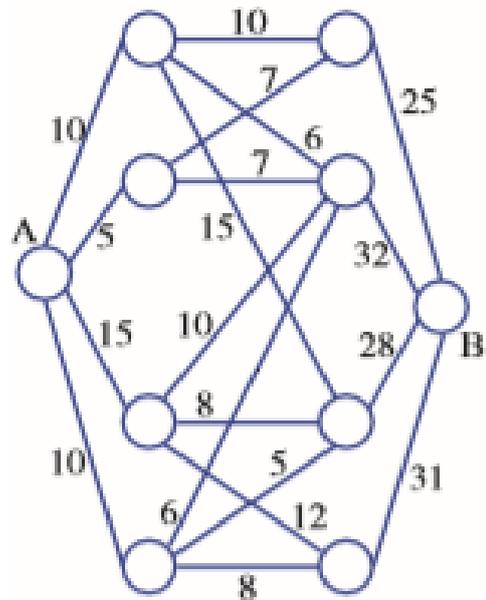
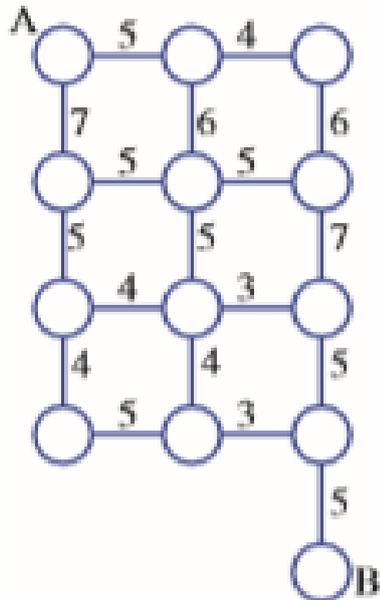
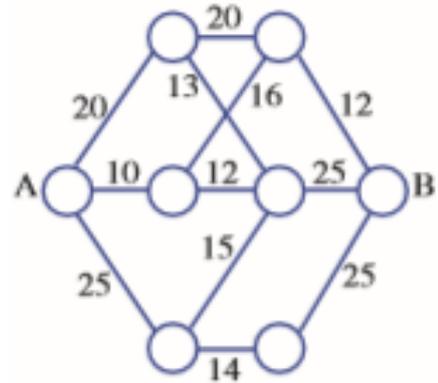
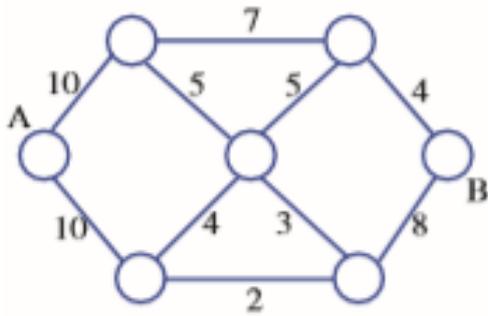
Write your answer.

The shortest path from A to P is A-B-F-J-K-L-P and is 14 minutes long.



Exercise 1

1. Find the length of the shortest path from A to B in each of the following networks. **Note:** Use a pencil and have an eraser handy.



2. The table shows the travelling times in minutes between towns which are connected directly to each other.
Note: The line indicates that towns are not connected directly to each other.

	Addisba	Bundong	Callop	Dilger	Eric
Addisba	0	50	20	25	-
Bundong	50	0	25	30	30
Callop	20	25	0	-	60
Dilger	25	30	-	0	70
Eric	-	30	60	70	0

- a. Draw a network to show the connection of the towns by these roads.

- b. Find the shortest travelling time between Addisba and Eric.

Critical path analysis

Activity charts and networks

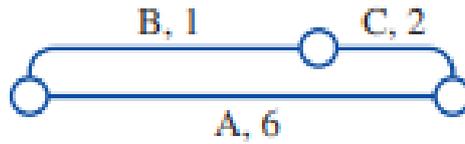
Not matter what we do in our lives, there are many tasks that we must fit into our daily schedule. If the daily tasks are not organised, we tend to run out of time or double-book ourselves. Similarly, operations such as major construction tasks must be efficiently planned so that the right people and materials are at the right place and at the right time. If one of these components is wrong, then time, and therefore money, are wasted.

To demonstrate the advantages of planning, we will use a simple example. Charlotte organises the things she needs to do, to allow time for the activities she wants to do. (These activities include getting up early for half an hour of tai chi and taking her dog for a run).

While getting ready for school in the morning, Charlotte is faced with the following problem. She has three tasks to do: downloading her email from the computer, reading the email and eating her breakfast. The first two tasks take 1 minute and 2 minutes respectively, while the last takes 6 minutes in all. Charlotte needs to complete all these tasks in 7 minutes. How might she accomplish this? Clearly, she needs to be able to do some tasks simultaneously. Although this seems like a simple problem, let us look at what might happen each minute.

Time	Activity	Activity
1st minute		Download email
2nd minute	Eat breakfast	
3rd minute	Eat breakfast	
4th minute	Eat breakfast	
5th minute	Eat breakfast	Read email
6th minute	Eat breakfast	Read email
7th minute	Eat breakfast	

Let's construct such a diagram for the problem. In the figure, the arcs of our network represent the three activities of downloading, reading, and eating. The left node represents the start of the activity, the right node represents the end of all activity, and the middle node indicates that the downloading must occur before the reading starts. In other words, downloading (B) is the immediate predecessor of reading (C). Another way of representing this information is in an activity chart.



Activity letter	Activity	Predecessor	Time (min)
A	Eat breakfast	-	6
B	Download email	-	1
C	Read email	B	2

This chart also shows that activity B (downloading) is the immediate predecessor of activity C (reading), and that the other two activities have no predecessors.

Let us now extend the activity chart to a more complex set of activities for Charlotte's morning routine.

Example 1

From the activity chart below, prepare a network diagram of Charlotte's morning schedule.

Activity letter	Activity	Predecessor	Time (min)
A	Prepare breakfast	-	4
B	Cook breakfast	A	2
C	Eat breakfast	B, E, G	6
D	Have shower	A	4
E	Get dressed	D	4
F	Brush teeth	C, H	2
G	Download email	A	1
H	Read email	B, E, G	2
Total time			25

Solution

Step 1: Begin the diagram by drawing the starting node.

Step 2: Examine the table looking for activities that have no predecessors. There must be at least one of these. Why?

This activity becomes the first arc and is labelled with its activity letter.

Step 3: List all activities for which A is the immediate predecessor.

Add a node to the end of the arc for activity A.

Activity A has no predecessors



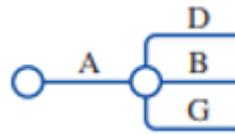
Activity B has A as an immediate predecessor.

Activity D has A as an immediate predecessor.

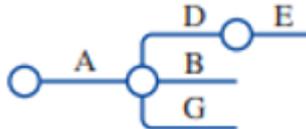
Activity G has A as an immediate predecessor.

Create one arc from this node for each of the listed activities. Label these arcs.

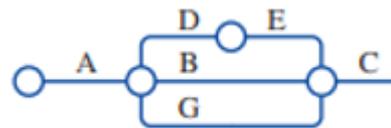
Note: The end node for each of these activities is not drawn until either you are certain that it is not the immediate predecessor of any later activities, or all activities have been completed.



Step 4: Repeat Step 3 for activity D. Since it is the *only* predecessor of activity E, this can be added to the diagram.

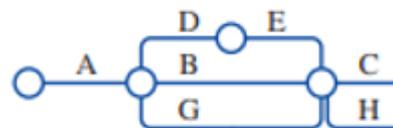


Step 5: Repeat Step 3 for activities B and G. they have no activities for which they are the only predecessors. Since activity C is preceded by all of B, G and E, join all the arcs at a single node.

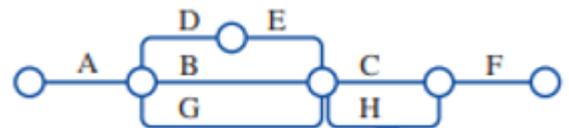


Add activity C after this joining node. *Note:* that activity H is also preceded by all of B, G and E but *not* by activity C.

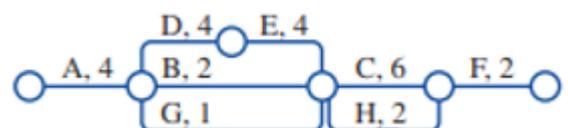
Step 6: Determine whether activity C and H are independent of each other. Since they are independent, activity H starts from the same node as activity C.



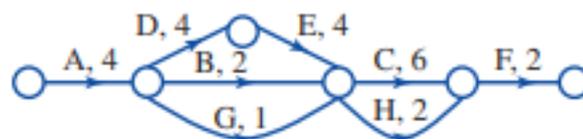
Step 7: The last activity is F, which has C and H as its immediate predecessors. Therefore, join C and H with a node, then add an arc for F. Since F is the final activity, also add the end node.



Step 8: Add the time required for each activity next to its letter.



An alternative network diagram is also shown. This diagram also indicates a direction: we start at task A, then perform tasks B, D and G ... and end up at task F. From the implied direction, these networks are called directed graphs or directed networks.



Now that the tasks have been reduced to a network diagram, we can use the diagram to help Charlotte reduce the total time spent on all these tasks. If all the tasks were spread out in a straight line, so no tasks completed at the same time, then her morning routine will take 25 minutes. The diagram shows that some of Charlotte's tasks can take place at the same time. Let's investigate the time savings available.

Forward scanning

By forward scanning through a network, we can calculate the earliest start times for each activity and the earliest completion time for the whole project. The **earliest start time (EST)** is the earliest that any activity can be started after all prior activities have been completed. The EST is determined by looking at all the previous activities, starting with the immediate predecessors, and working back to the start of the project. An activity can start no earlier than the completion of such predecessors. Obviously, the EST for the first activity is 0.

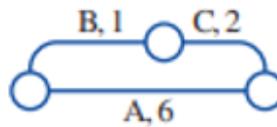
To determine the time saving, first determine the earliest start time for each activity. For simplicity, we will return to the initial three tasks with which Charlotte was faced: downloading and reading her email and eating her breakfast.

Example 2

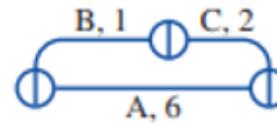
Use forward scanning to determine the earliest completion time for Charlotte's initial three tasks from the previous example.

Solution

Step 1: Begin with the network diagram.

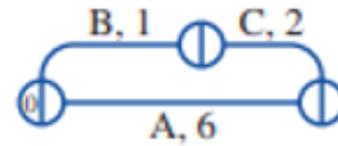


Step 2: Separate nodes into two halves.

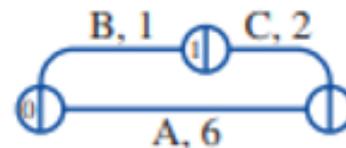


Step 3: The earliest start time for each node is entered in the left-hand side of the node. Nodes with no immediate predecessors are given the value of zero.

As activities B and A have no immediate predecessor then their earliest start time is zero.

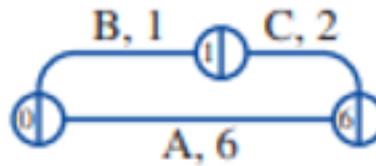


Step 4: Move to another node and enter the earliest start time in the left-hand side. In the case of activity C, it must wait one minute while its immediate predecessor B, is completed.



Step 5: The last node's earliest start time is entered. When more than one arc joins at a node, then the earliest start time is the largest value of the paths to this node. This is because all tasks along these paths must be completed before the job is finished.

Path B-C = 1 + 2 = 3 minutes
 Path A = 6 minutes



There are two paths converging at the final node. The top path takes 3 minutes to complete and the bottom path, 6 minutes. The larger value is entered in the node.

Step 6: The earliest completion time is the value in the node. All tasks can be completed in 6 minutes.

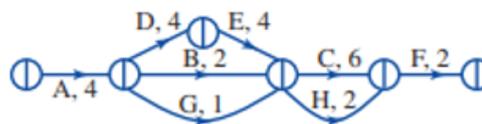
It is important for anybody planning many tasks to know which tasks can be delayed and which tasks must be completed immediately. In the worked example above, the eating must be commenced immediately if the 6-minute time is to be attained, whereas downloading the email could be delayed three minutes and still allow enough time for it to be read while Charlotte is eating.

Example 3

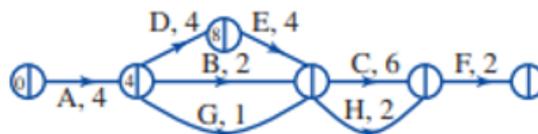
Use all the activities listed in Charlotte's morning routine in the first example, find the earliest completion time and hence identify those tasks that may be delayed without extending the completion time.

Solution

Step 1: Draw a directed network with split circles at each node.

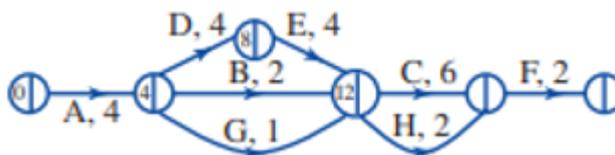


Step 2: Begin forward scanning. The earliest start time for the first three nodes in the path can be entered immediately.



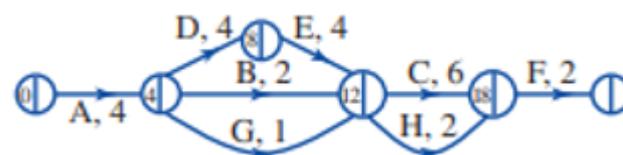
Step 3: Calculate the time values for the paths to the fourth node. Enter the largest value into the left-hand side of the node.

A-D-E = 4 + 4 + 4 = 12 minutes.
 A-B = 4 + 2 = 6 minutes.
 A-G = 4 + 1 = 5 minutes.



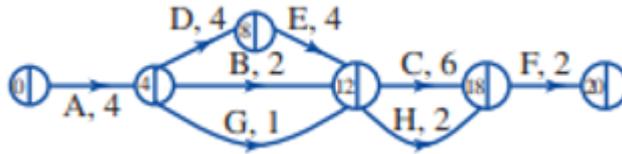
Step 4: Repeat Step 3 for the next node. *Note:* calculations begin by using the time from the previous node (12 minutes),

A-E-C = 12 + 6 = 18 minutes.
 A-E-H = 12 + 2 = 14 minutes.



Step 5: There is only one path to the last activity (F). Add its time requirement to that of the previous node (18 minutes).

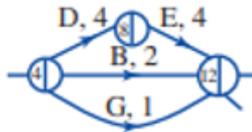
$$A-C-F = 18 + 2 = 20 \text{ minutes}$$



Step 6: The time in the last node indicates the earliest completion time.

Earliest completion time is 20 minutes.

Step 7: Identify sections of the network where there was a choice of paths. There are two such sections of the network. Examine the first one (at the 4th node).



Step 8: List and total the time for each path through this section of the network. The largest value indicates the path that cannot be delayed.

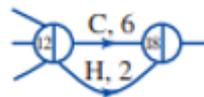
$$D-E = 4 + 4 = 8 \text{ minutes}$$

$$B = 2 \text{ minutes}$$

$$G = 1 \text{ minute}$$

Paths B and G can be delayed

Step 9: Repeat Step 8 for the next section identified in Step 7.



$$C = 6 \text{ minutes}$$

$$H = 2 \text{ minutes}$$

H can be delayed

The path through the network which follows those activities that cannot be delayed without causing the entire project to be delayed is called the **critical path**.

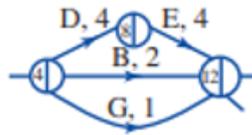
Therefore, the critical path for the activities listed in Charlotte's morning routine would be A – D – E – C – F. It is easily seen that this path takes the longest time (20 minutes).

Float time and latest start time

Float time is the maximum time an activity can be deferred without delaying the entire project. The latest start time for such activities is defined as the latest time they can start without delaying the project.

Example 4

Work out the float time for activities B and G in Example 3 and identify the latest starting time for these activities.



Solution

Step 1: List the alternative paths for the section containing activities B and G and the times for these alternatives.

D-E = 4 + 4 = 8 minutes
 B = 2 minutes
 G = 1 minute

Step 2: Subtract the smaller times separately from the maximum time.

Float time for activity B = 8 - 2 = 6 minutes
 Float time for activity G = 8 - 1 = 7 minutes

Step 3: Look up the **earliest completion time** for the activity on the critical path and subtract the activities' times.

D-E is on the critical path.
 Earliest completion time = 12 minutes
 Latest start time for activity B = 12 - 2 = 10 minutes
 Latest start time for activity G = 12 - 1 = 11 minutes

The float times indicate the amount of time of which these activities can be deferred without delaying the completion of all tasks. Furthermore, activity B could begin up to 6 minutes (4 + 6) after the start of the critical activity (D) while G could begin up to 7 minutes (4 + 7) after the same critical activity (D).

Exercise 2

1. For each of the activity charts below, prepare a network diagram
 - a.

Activity	Immediate predecessor
A	—
B	—
C	A

- b.

Activity	Immediate predecessor
D	—
E	D
F	D
G	E, F

c.

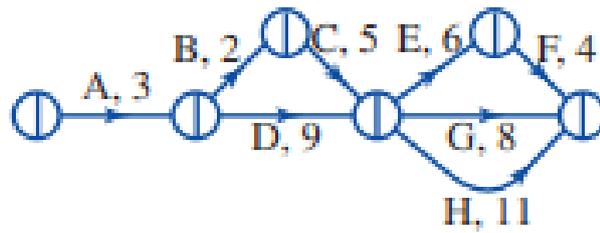
Activity	Immediate predecessor
A	—
B	A
C	A
D	C
E	B
F	B
G	F
H	D, E, G
J	D, E, G
I	J, H

2. When a personal computer is being assembled, the following processes must be performed.

Activity letter	Activity	Predecessor	Time (min)
A	Install memory board	—	2
B	Test hard drive	A	20
C	Install hard drive	B, E	4
D	Install I/O ports	A	5
E	Install CD-ROM	D	3
F	Test CD-ROM	E	5
G	Install operating system	C, F	10
H	Test assembled computer	G	12
Total time			61

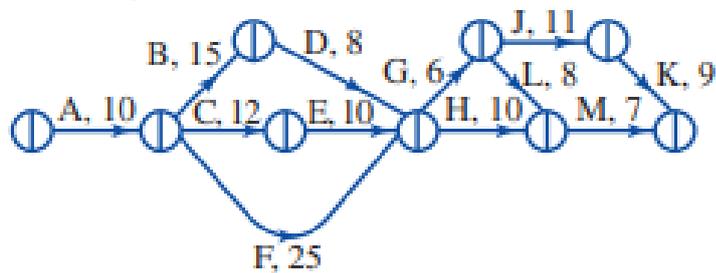
- Construct a network diagram.
- Determine the minimum time in which all tasks could be completed.

3. From the diagram:



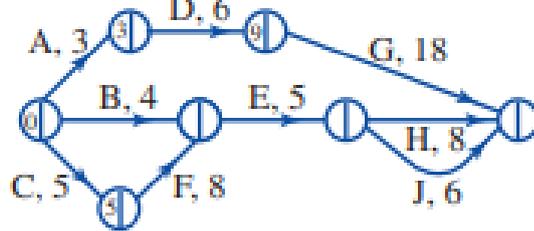
- a. Use forward scanning to determine the earliest completion time.
- b. Identify tasks that may be delayed without increasing the earliest completion time.

4. From the diagram:



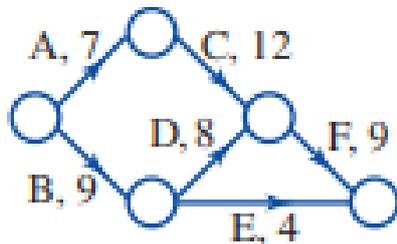
- a. Find the earliest start time for each node in the network.
- b. Find the earliest completion time for the project.
- c. Produce an activity chart.

5. From the diagram:



- a. Find the critical path.
- b. Determine which activities have float time and calculate their float times.
- c. Determine the latest start time for all non-critical activities.
- d. Provide an activity chart.

6. From the diagram:



The times are shown in minutes.

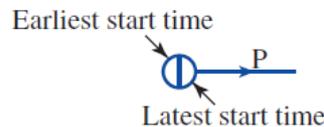
- a. Which of the following statements is true?
 - i. Activity A is an immediate predecessor of F
 - ii. Activity D is an immediate predecessor of F
 - iii. Activity F must be done before D
 - iv. Activity F must be done before E
- b. The minimum time taken to complete all activities is:
 - i. 19 minutes
 - ii. 21 minutes
 - iii. 23 minutes
 - iv. 28 minutes
- c. Determine the critical path for the network.

Critical path analysis with forward and backward scanning

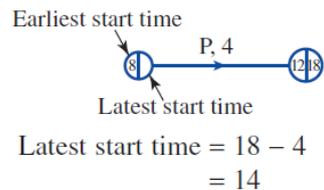
With more-complex projects requiring the coordination of many activities, it is necessary to record more information on the network diagrams and to display the information using charts. In the previous book, the float times and the critical path were worked out using somewhat informal methods. In this section, a more formal method will be shown to enable float times to be calculated and the critical path to be determined. This method involves backward scanning.

Backward scanning

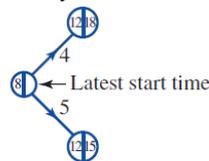
To complete a critical path analysis, a procedure called **backward scanning** must be performed. In forward scanning, we record the earliest start time for any activity in the left-hand side of each node; in backward scanning we record the **latest** start time in the right-hand side of each node – that is, the latest time that this activity can start without delaying the project.



Backwards scanning starts at the end node and moves backward through the network subtracting the time of each arc from the earliest start time of each succeeding node. When two or more paths are followed back to the same node, the smallest difference is recorded.



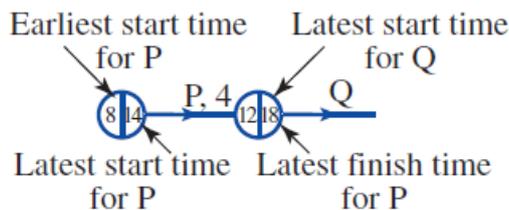
The results of each backward scanning step yield the latest start time for each activity. Latest start time is the latest time an activity can start without delaying the project.



$$\begin{aligned} \text{Latest start time} &= 15 - 5 \text{ (because it is smaller than } 18 - 4) \\ &= 10 \end{aligned}$$

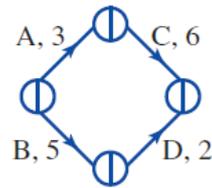
Latest finish time (LFT) for an activity is equal to the latest start time for the following activity. **Float time** is the maximum time that an activity can be delayed without delaying a subsequent activity on the critical path and thus affecting the earliest completion time. From the definitions, there is a relationship between float time and the other quantities, namely:

$$\text{Float time} = \text{latest finish time} - \text{earliest start time} - \text{activity time}$$



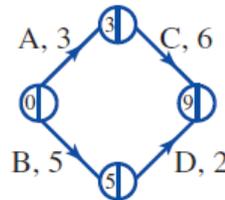
Example

The network diagram has been constructed for a project manager. Use forward and backward scanning to clearly display the critical path and to list any float times.



Solution

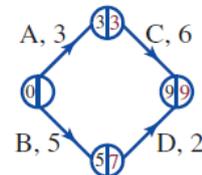
Step 1: Forward scan through the network and record the earliest start time for each activity in the left-hand side of the appropriate node.



Step 2: Begin backward scanning. Start at the end node and trace backwards along all paths from this node.

Along path C: $9 - 6 = 3$
 Along path D: $9 - 2 = 7$

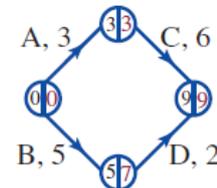
Subtract the times of the activities along each path from the earliest completion time and record the value in the right-hand side of the previous node. These values are the latest start times for the activities along the path.



Latest start time for C = 3
 Latest start time for D = 7

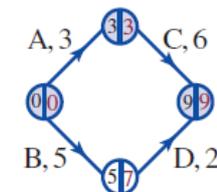
Step 3: Repeat the process backwards through the diagram. Where two (or more) paths come together (activities A and B), record the *smaller* value in the right-hand side of the node.

Along path A: $3 - 3 = 0$
 Along path B: $7 - 5 = 2$
 Smallest value = 0



Step 4: The critical path can now be clearly identified. It is the path that has the same numbers in both the left and right sides of any node. Remember to include *all* such nodes in the critical path.

Critical path is shaded (see digital copy).



Step 5: Float times are calculated now. Construct a table with the headings shown.

Activity	Activity time	Earliest start time	Latest finish time	Float time
A	3	0	3	0
B	5	0	7	2
C	6	3	9	0
D	2	5	9	2

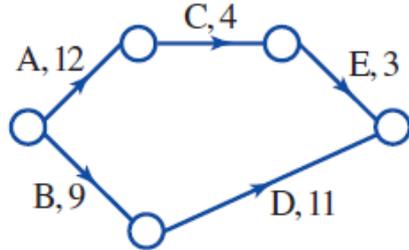
Record the times from the left-hand side of the nodes in the earliest start times (EST) column, the times in the right-hand side of the nodes in the latest finish times (LFT) column as well as the activity times (T). Calculate float times using the equation:
 Float = LFT – EST – T

For activity D: Float = $9 - 5 - 2 = 2$
 For activity C: Float = $9 - 3 - 6 = 0$
 For activity B: Float = $7 - 0 - 5 = 2$
 For activity A: Float = $3 - 0 - 3 = 0$

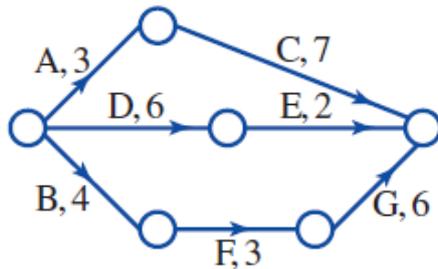
In this example the float times are also the differences between the corresponding times in the nodes. This is not the rule in the general case.

Exercise 3

- For the network diagram, use forward and backward scanning to clearly display the critical path and to list any float times. Time is in minutes.

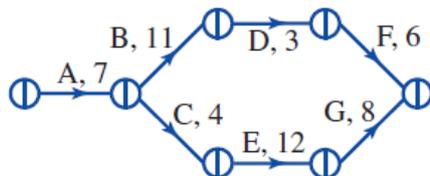


- For the network diagram, use forward and backward scanning to clearly display the critical path and to list any float times for non-critical activities. Time is in hours.



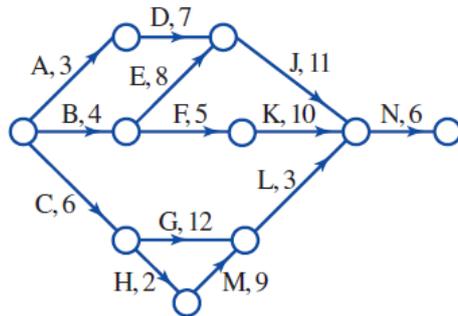
- Complete the figure by forward and backward scanning and hence:
 - Determine the earliest completion time
 - Indicate the critical path
 - The float time for activity D
 - The latest start time for activity D

Note: Time are in days



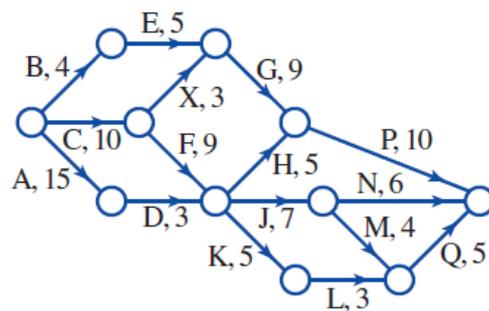
4. The manufacturing of bicycles can be considered as a 7-step process:
- A — Collect all the parts — 12 minutes
 - B — Paint frame — 35 minutes (requires A to be completed first)
 - C — Assemble brakes — 16 minutes (requires A to be completed first)
 - D — Assemble gears — 20 minutes (requires B to be completed first)
 - E — Install brakes — 12 minutes (requires C to be completed first)
 - F — Install seat — 5 minutes (requires C to be completed first)
 - G — Final assembly — 18 minutes (requires D and E to be completed first)
- a. Construct an activity chart.
 - b. Construct a network diagram.
 - c. Determine the earliest completion time using forward and backward scanning.
 - d. Determine the critical path.
 - e. What activities have float times?

5. From the network diagram:



- Complete a forward scan and determine the earliest completion time
- Complete a backward scan and determine the critical path

6. From the network diagram



- Complete a forward scan and determine the earliest completion time
- Complete a backward scan and determine the critical path
- Determine the float time for activity X

2022 MA4 Week 11 and 12 Investigation

Task 1 Prepare a network diagram for the following activity chart.

Activity	Immediate predecessor
N	—
O	N
P	O, T
Q	P
R	—
S	N
T	S, Y
U	O, T
V	O, T
W	V
X	Y
Y	R
Z	U, X

Task 2 Create an activity chart of your own

Complete an activity chart for a routine task or project such as baking a cake, making your favourite dinner, writing and submitting an essay, completing a maths brief or an art project. Choose a task that requires at least four activities including several predecessors. Include the following headings in your table.

Activity letter	Activity	Predecessor	Time
-----------------	----------	-------------	------

MARKING RUBRIC**Week 11 and 12****NAME:**

CRITERIA	EXPECTATIONS	POSS	MULT	GIVEN	TOTAL
Practical	Student completes practical work, including exercises and any Mathspace and/or other tasks, of the brief to an acceptable standard set by the teacher.	2	3		/6
Investigation Task	Student completes the investigation task of the week to an acceptable standard set by the teacher.	2	2		/4
Communication and Reasoning	Student responses are accurate and appropriate in presentation of mathematical ideas in different contexts, with clear and logical working out shown.	4	-		/4
Knowledge and Application	Student submitted work selects and applies appropriate mathematical modelling and problem-solving techniques to solve practical problems and demonstrates proficiency in the use of mathematical facts, techniques, and formulae.	4	-		/4
	Submission Guidelines				
Timeliness	Student submits the practical work, including exercises and any Mathspace and/or other tasks, and investigation by the set deadline. See scoring guidelines for specific details.	2	-		/2
				FINAL	/20

Student Reflection: How did you go with this week's work? What was interesting? What did you find easy? What do you need to work on?