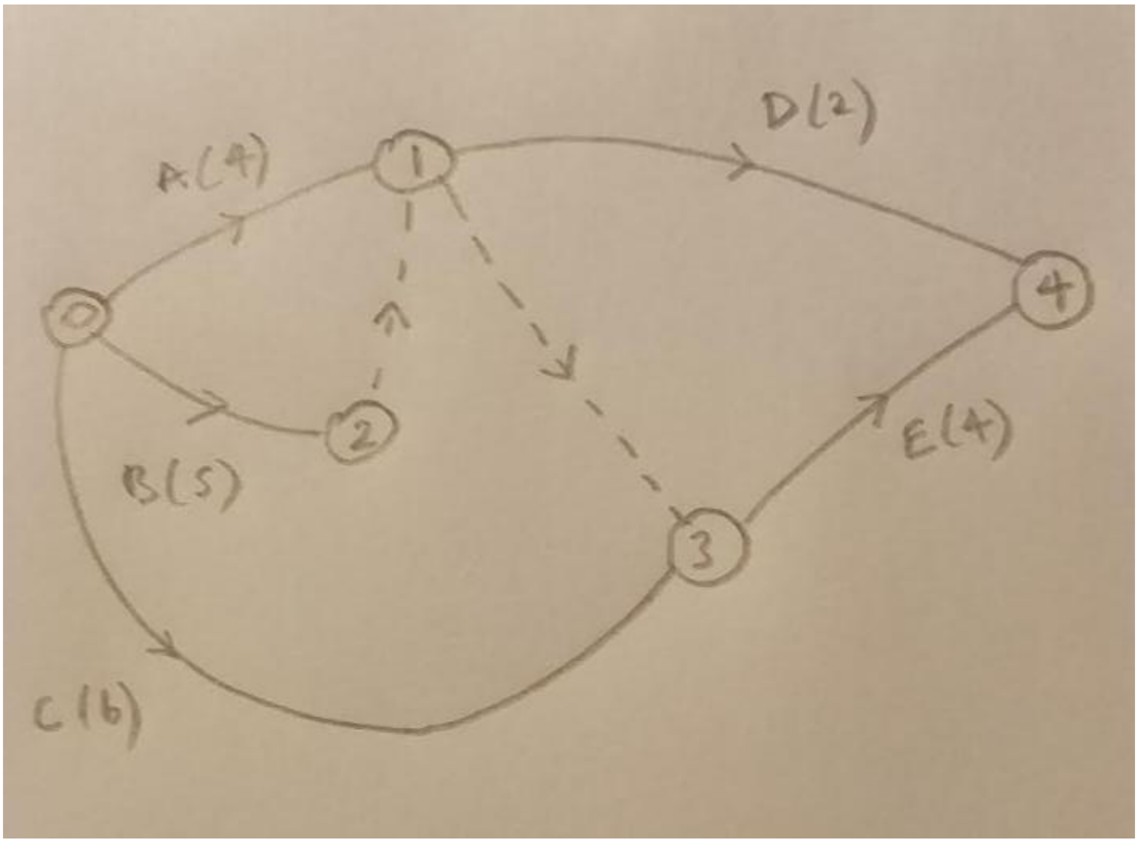
Critical Path Analysis - Part 1 (12 pages; 17/1/26)

(1) The starting point is the **precedence table** (aka dependence table)

Example

Activity	Duration	Immediately preceding activities
A	4	-
B	5	-
C	6	-
D	2	A,B
E	4	A,B,C

(2) The **activity network** can then be constructed from this:



(3) In the case of Dijkstra's Algorithm, we are only concerned with one route in a network (the shortest one). For Critical Path Analysis however, the various branches of the network are in action at the same time. They could be various processes taking place in a factory that go towards completing a particular product.

This can be represented by an **activity-on-arc** network. The nodes represent the starting points of activities. The first node is sometimes called the **source node**, and the final node is sometimes called the **sink node**.

[An alternative method uses an **activity-on-node** network. An example of this is given later. The AQA exams are based on this method.]

(4) Dummy activities

These are the dashed arcs in the network above. They have zero duration, and can have two purposes:

(i) The purpose of the dummy activity leading from node 2 to node 1 is to be able to distinguish between activities A and B – otherwise they would both be labelled (by a computer programme say) as (0,1) (ie starting at node 0 and ending at node 1).

(ii) The purpose of the dummy activity leading from node 1 to node 3 is to record the fact that activity E (or more generally any activity leading from node 3) is directly* dependent on activities A and B, as well as activity C (ie these activities need to have taken place before E can start). (Whereas activity D is only dependent on activities A and B.)

* Note that only direct dependencies are shown in the Precedence Table. So if activity B is recorded as depending on activity A, and activity C is recorded as depending on activity B, then it will of course be true that C can't start until both A and B have taken place.

Exercise: Construct an activity network for the following part of a precedence table:

	depends on
A	–
B	–
C	–
D	A, B
E	A, C

Solution

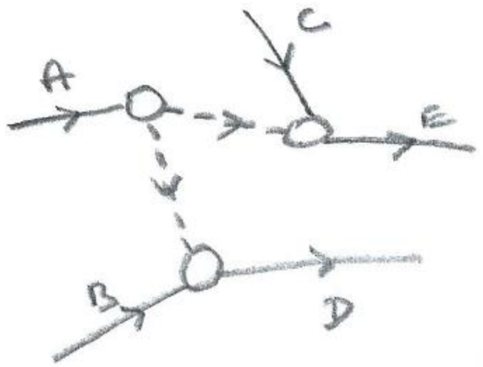
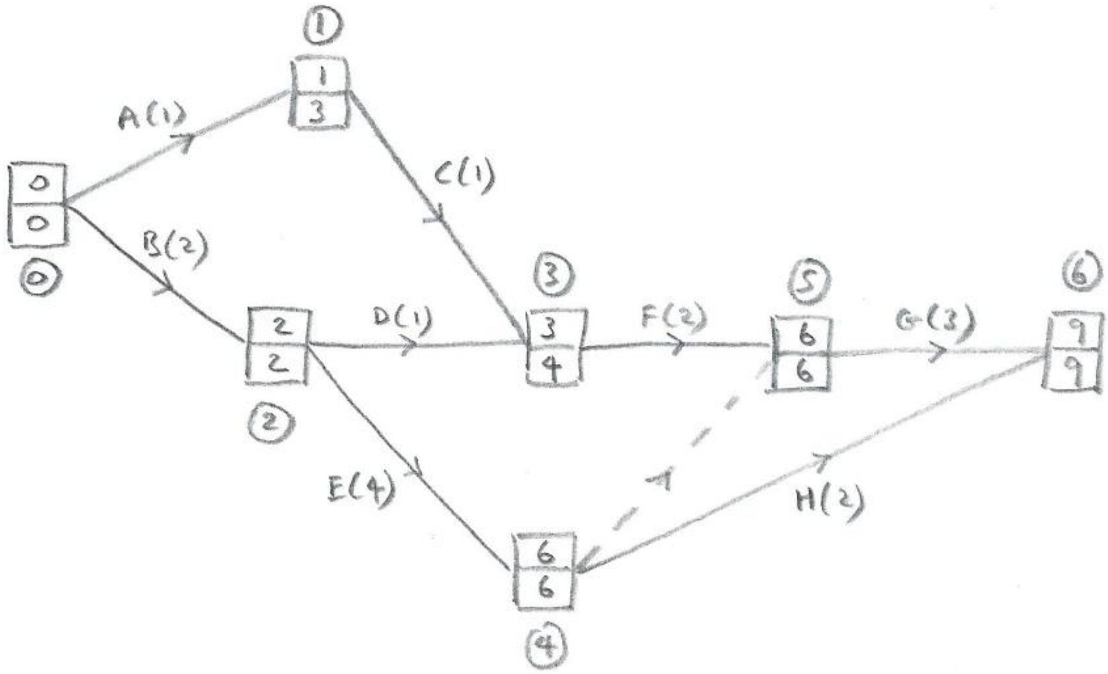


Fig. 1

(5) Earliest/latest event times

Activity	Duration	Immediately preceding activities
A	1	-
B	2	-
C	1	A
D	1	B
E	4	B
F	2	C,D
G	3	E,F
H	2	E



The above example shows the **earliest and latest event times** for each node. (The 1st number in the box is the earliest event time, and the 2nd is the latest event time.)

[These are sometimes called the early and late event times.]

The earliest event times (EETs) are determined first.

For node 1, the earliest event time is the earliest time (measured from the start of the process) that activity C can start. As the duration of A is 1, this EET is 1.

For node 2, the EET is the earliest time that activities D and E can start.

For node 3, the EET is the earliest time that activity F can start, once activites C and D have both been completed. The earliest time that activity C can be completed is the EET for node 1, plus the duration of C: thus, $1 + 1 = 2$. The earliest time that activity D can be completed is the EET for node 2, plus the duration of D: thus, $2 + 1 = 3$. So the EET for node 3 is the greater of 2 and 3; ie 3.

The process of establishing the EETs is known as a **forward pass**.

The EET of the final node is the **critical time** or **minimum completion time** for the whole operation.

The latest event time (LET) for a particular node is the latest time (again, measured from the start of the process) at which at least one of the activities leading from that node must start, in order that the critical time for the whole process can still be achieved.

The LETs are determined by setting the LET for the final node equal to its EET (as this is the critical time for the whole process), and working back from the final node:

Thus, for node 5, activity G must start by time 6 (measured from the start), in order for the whole process to be completed by time 9 (as the duration of G is 3).

For node 4, activity H need only start by time 7, in order for the whole process to be completed by time 9 (as the duration of H is 2). However, because there is also a dummy activity leading from node 4, and the LET for node 5 is 6, the LET for node 4 has to be the smaller of 7 and 6; ie 6 (in order for activity G to be able to start by time 6).

The process of establishing the LETs is known as a **backward pass.**

Note that, for EETs we are finding the **largest** of a number of completion times of activities, whilst for the LETs we are finding the **smallest** of a number of start times of activities.

(6) Earliest start and latest finish times

Suppose activity (i, j) starts at node i and finishes at node j. Then the **earliest start time** for this activity is $EET(i)$, and the **latest finish time** is $LET(j)$. (These terms sound very similar to "earliest event time" and "latest event time", but we are now considering an activity, whereas the events relate to the points where activities start and end.)

These expressions can be used to define the total float (as seen below), and are also used in the activity-on-node method for constructing the activity network, as described next.

(7) Activity-on-node method

As the title suggests, each activity is represented by a node, which shows the following information for the activity:

- the earliest start time

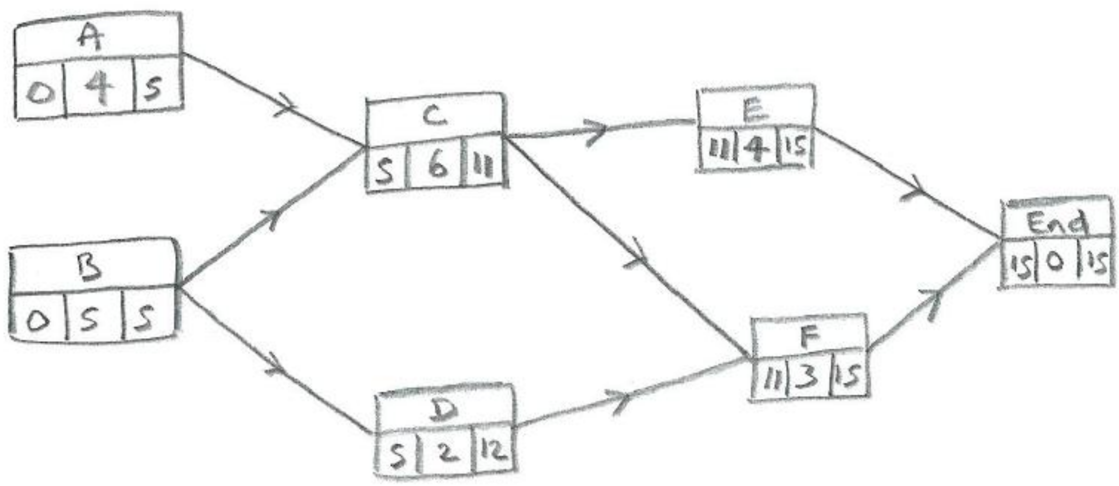
- the duration

- the latest finish time

Example

Activity	Duration	Immediately preceding activities
A	4	-
B	5	-
C	6	A,B
D	2	B
E	4	C
F	3	C,D

Activity-on-node network



Notes

(i) The 3 figures in the box are: earliest start time, duration, and latest finish time.

(ii) An ‘end activity’ is included.

(ii) A forward pass is carried out to determine the earliest start times, and a backward pass to determine the latest finish times.

For example, the EST for activity F is

$\max(5+6\,[C], 5+2[D]) = 11$,

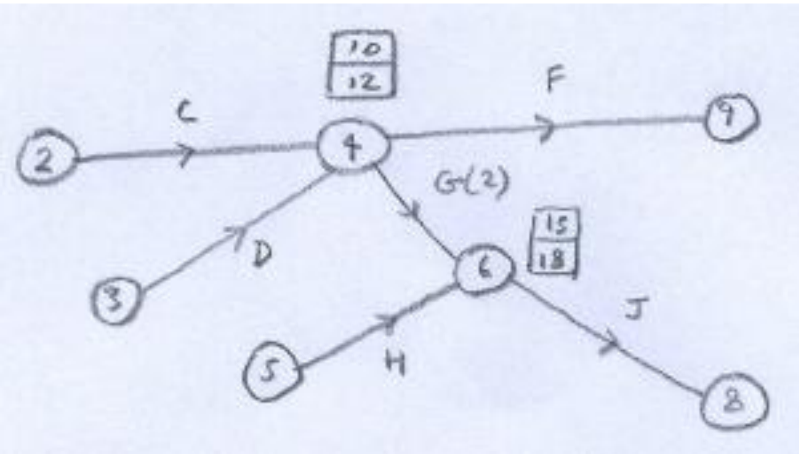
and the LFT for activity C is $\min(15-4[E], 15-2[F]) = 11$.

Note: The LFTs for E and F are automatically the same as for the ‘end activity’.

(iii) No dummy activities are needed for this method.

[Resuming activity-on-arc]

(8) Floats



Referring to the diagram above:

10 is the earliest time by which C and D can both be completed (and hence the earliest time by which F or G could start).

12 is the latest time by which both C and D need to have been completed, in order that F or G (as necessary) can start early enough for the critical time to be achieved.

$18 - 10 = 8$ is the maximum time available for G, if given priority over other activities for any spare time.

2 is the (minimum) duration needed for G.

The **total float** for G $= 8 - 2 = 6$ (ie the maximum spare time available).

The **independent float** for G $= (15 - 12) - 2 = 1$ (the float that G can be assured of, without relying on the other activities).

The independent float is set equal to zero, if it would otherwise be negative.

The **interfering float** is $6 - 1 = 5$ (ie total float $-$ independent float).

In general, the total float is

$LET(j) - EET(i) -$ duration of activity,

where i represents the 1st node and j represents the 2nd

(or latest finish time $-$ earliest start time $-$duration of activity)

and the independent float is:

$Max\{EET(j) - LET(i) -$ duration of activity, $0\}$

(9) Critical Activities

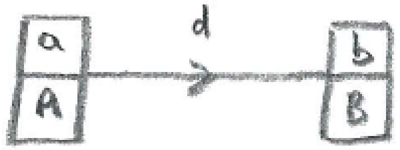
A **critical activity** is where there is no margin for delay, and so where the total float is zero.

Thus $LET(j) - EET(i) =$ duration of activity

Also, it can be shown that

$LET(i) = EET(i)$ and $LET(j) = EET(j)$.

Proof



Referring to the diagram above,

Critical activity $\Rightarrow B - a = d$ (1)

Also $b \geq a + d$ (2) & $A \leq B - d$ (3)

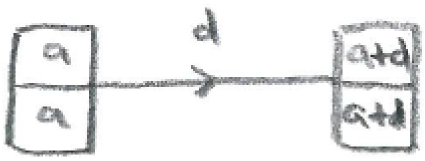
and of course $a \leq A$ (4) & $b \leq B$ (5)

Then , from (1) & (3), $A \leq (d + a) - d = a$,

so that $A \leq a$ & $a \leq A$, from (4), and hence $a = A$

Also, from (1) & (2), $b \geq B$ and since $b \leq B$ from (5), it follows that $b = B$.

So a critical activity can always be represented as shown below.



(10) Critical Paths

A **critical path** is one made up entirely of critical activities, and its length is the critical time. (It is sometimes noted that this length is the longest length of a path in the network: paths involving non-critical activities will be shorter.)

As we have seen, critical paths can be identified by looking for the nodes where $LET = EET$, and joining the ones where

$LET(j) - EET(i) = \text{duration of activity}$

(Note that there may be activities for which $LET(i) = EET(i)$ and $LET(j) = EET(j)$, but where $LET(j) - EET(i) >$ duration of activity, and these are obviously not critical activities.)

There may be more than one critical path. If so, they will all have the same length.

For the network in (4), the only critical path is $BB'CE$.