Software Project Management

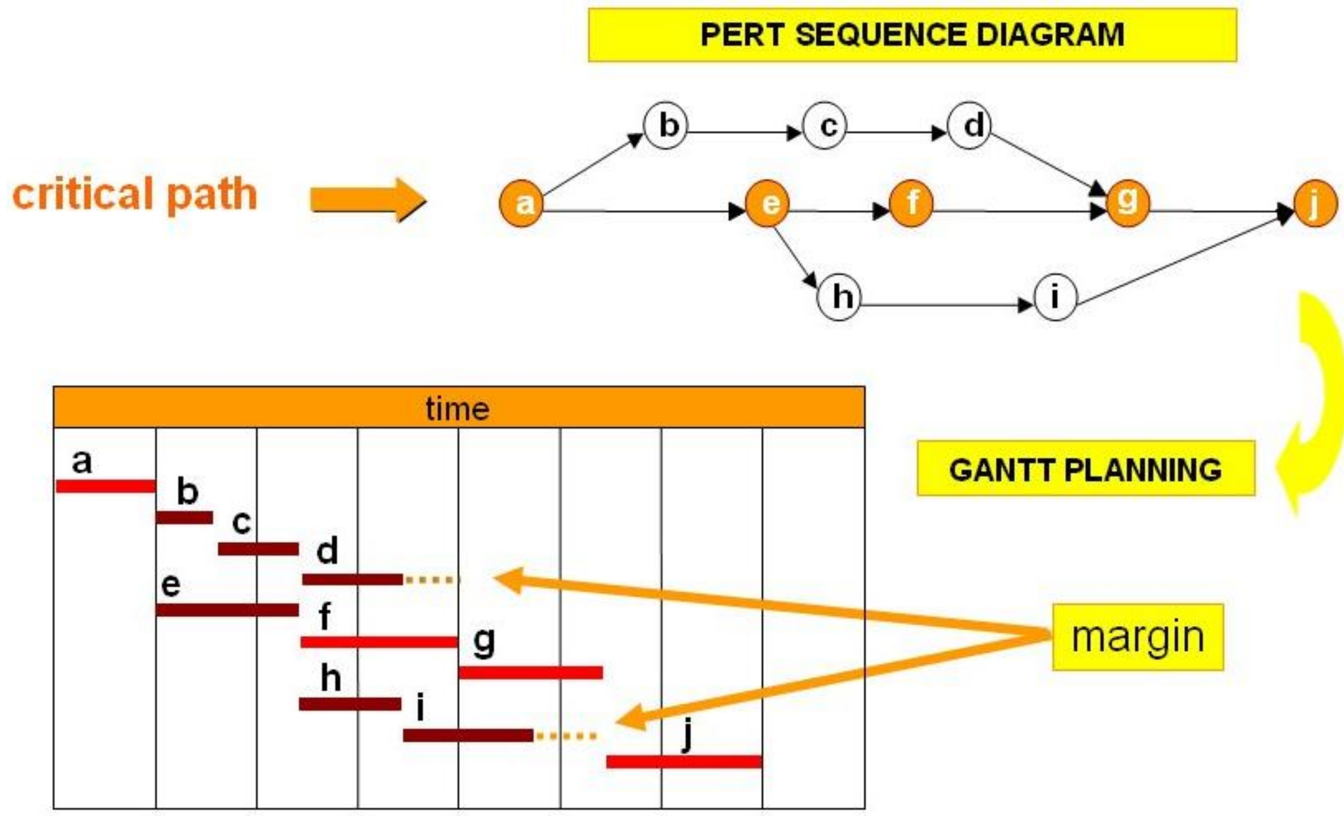
Project Scheduling –

- ✓ **Gantt chart**
- ✓ **PERT diagram**
- ✓ **CPM**

Project Work plan (/scheduling)

- It is a dynamic document that defines the list of tasks and important information about each task.
- A project plan includes: start date, completion date, estimated duration, actual duration, precedence information, deliverables, completion status, required resources, key milestones, staffing plan, etc.
- Common Tools to produce a Workplan
  - Gantt Chart - is a type of bar chart that illustrates a project schedule.
  - PERT Diagrams - useful tool to identify critical tasks.

Gantt Chart & Pert Diagrams Examples



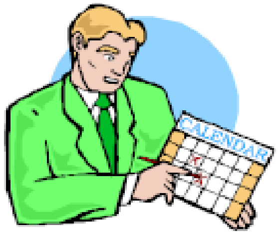
Reading: (Kendall&Kendall, chapter 3), (Dennis&Wixom, chapters 2,3)

Lecture – PERT Diagrams & CPM

- Project Scheduling
- Project Scheduling Objectives
- Phases of Project Scheduling
- PERT Diagrams & Dummy Activities
- CPM – Critical Path Method

Project Scheduling

- It is part of project management within the *Planning* phase of the Systems Development Life Cycle.
- Project Scheduling: Allocate resources to execute all activities in the project.



- Project: Set of activities or tasks with a clear *beginning* and *ending* points. The amount of available resources (time, personnel and budget) to carry out the activities is usually limited.

PROJECT SCHEDULING

- **Objectives:**
  - Establish beginning, ending and duration of each activity in the project.
  - Calculate overall completion time of the project given the amount of usually limited resources.
  - Determine the critical path and its duration.
  - Determine the slack time for all non-critical activities and the whole project.
  - Guide the allocation of resources other than time such as staff and budget.

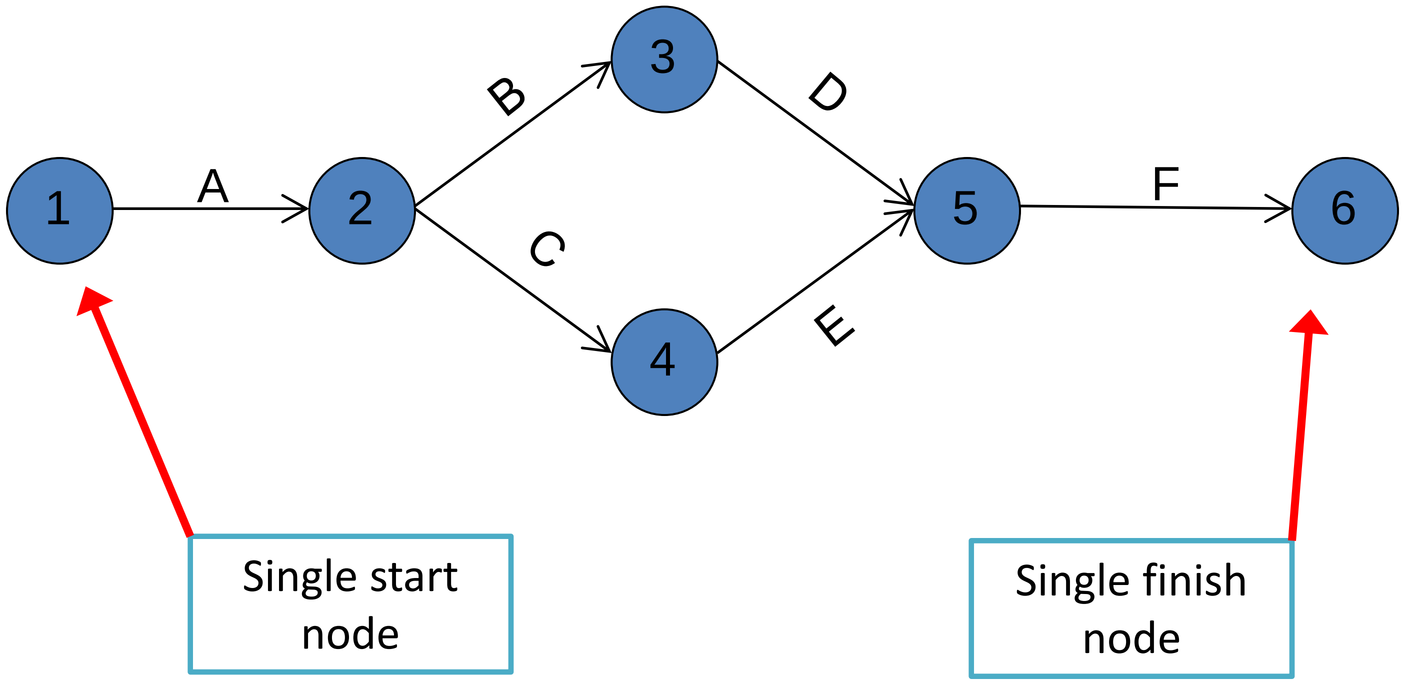
Project Scheduling

- Phases:
  - Define activities or tasks according to the project objectives.
  - A task is an individual unit of work with a clear beginning and a clear end.
  - Identify precedence relationships or dependencies
  - Estimate time required to complete each task.
  - Draw an activity-on-arrow PERT diagram inserting dummy activities if required.
  - Apply CPM to calculate *earliest* and *latest starting times*, *earliest* and *latest completion times, slack* times, *critical path* etc.
  - Construct a GANTT chart.
  - Reallocate resources and resolve if necessary.
  - Continuously monitor/revise the time estimates along the project duration.

PERT Diagrams

- **Program Evaluation and Review Technique**
- It is a network model that allows for randomness in activity completion times.
- Tool used to control the length of projects.
- PERT was developed in the late 1950’s for the US Navy’s Polaris Project.
- First used as a management tool for military projects
- Adapted as an educational tool for business managers
- It has the potential to reduce both the time and cost required to complete a project.

PERT Diagrams



PERT Diagrams

- Activity-on-node diagrams:
  - Maybe more than one single start and end node
  - Nodes represent activities
  - Arrows indicate precedence

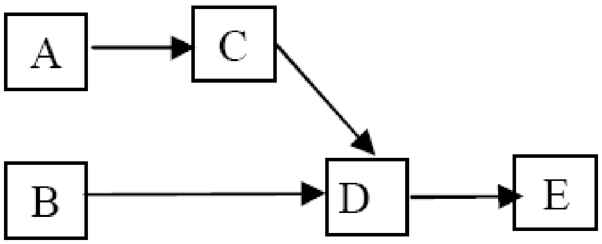
- Activity-on-arrow diagrams:
  - One single start and one single end node
  - Arrows represent activities
  - Nodes indicate beginning/end of activities

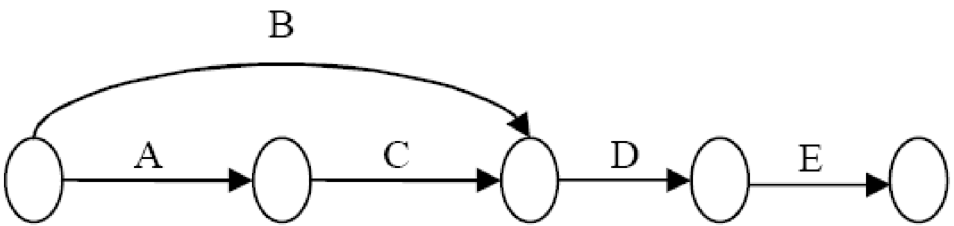
PERT DIAGRAMS

Example of PERT diagrams:

Task	Precedence
A	
B	
C	A
D	B,C
E	D

Activity-on-node

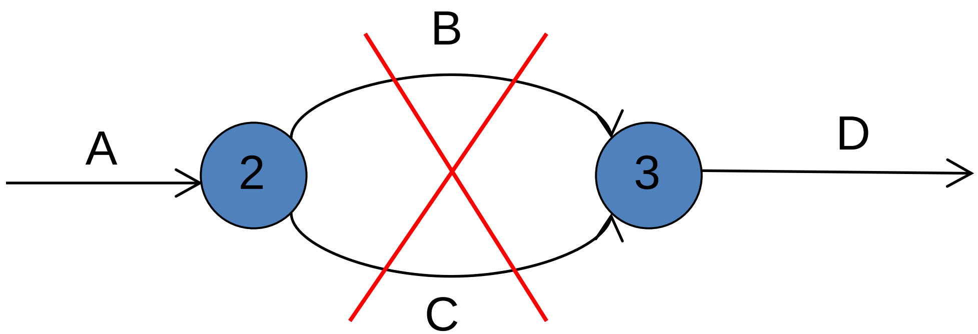




Activity-on-arrow

PERT Diagrams

- Some basic rules for Activity –on-arrow:
  - Tasks are represented as arrows
  - Nodes represent the start and finish points of tasks
  - There is only one overall start node
  - There is only one overall finish node
  - Two tasks cannot share the same start and end node.



Tasks B & C share the same start and end node

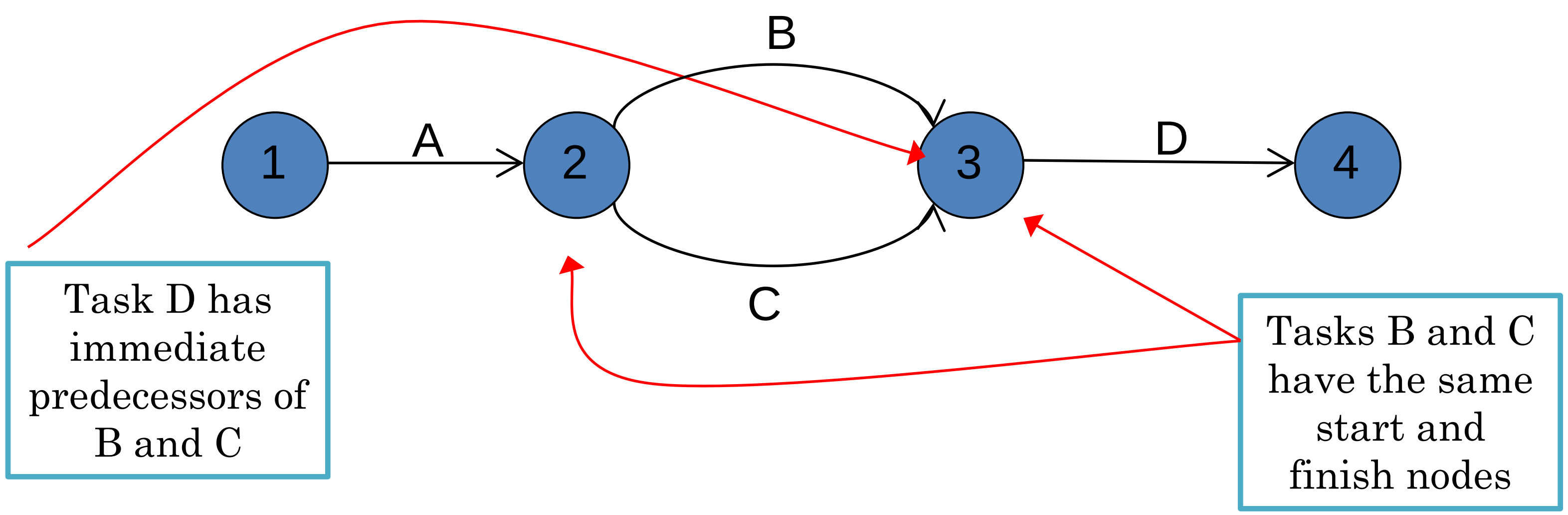
Dummy Activities

- Sometimes it is necessary to insert dummy activities (duration zero) in order to maintain the clarity of the diagram and the precedence relationships between activities.
- In activity-on-arrow PERT diagrams, each activity must be uniquely identifiable by its start and end nodes.
- However, sometimes multiple tasks have the same predecessors and successors.

Dummy Activities

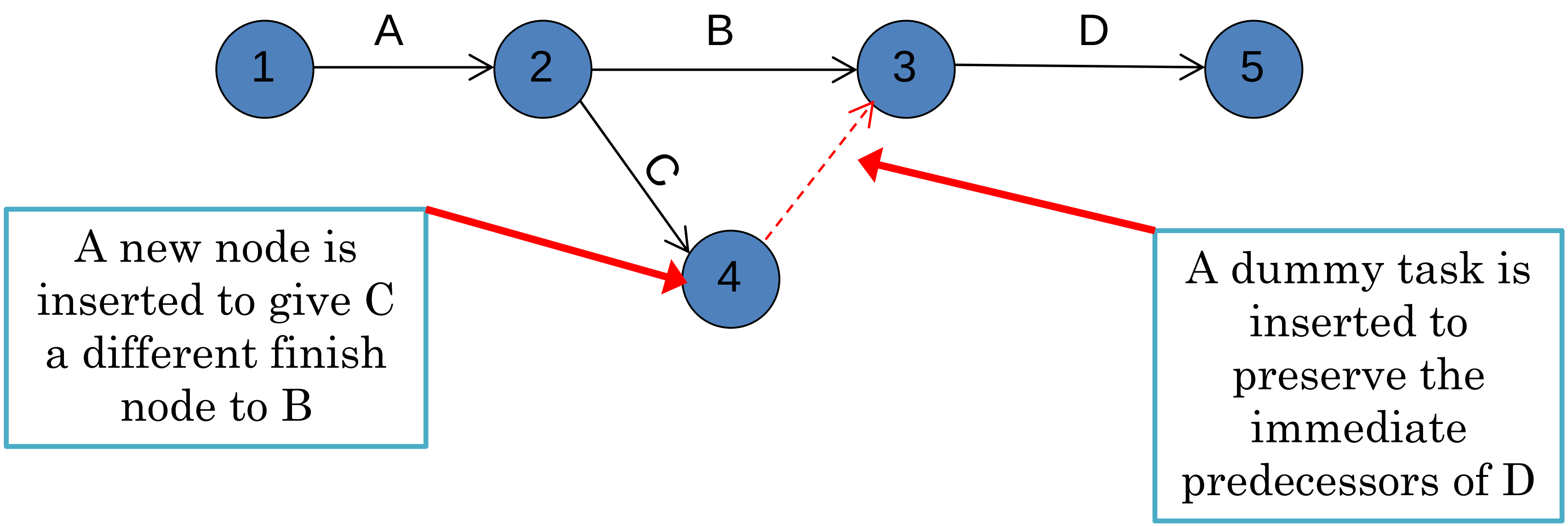
Case One

- A task should be uniquely identifiable from its start node and finish node
  - ✓ *This means that two or more tasks cannot share the same start and finish nodes*



Dummy Activities

- Inserting a *dummy activity* can ensure that multiple tasks have different successors.

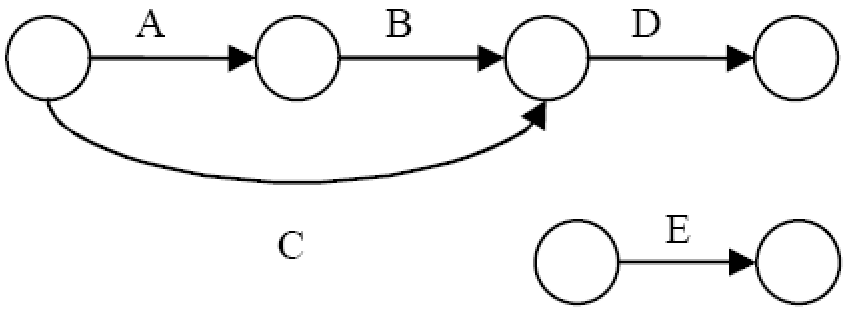


Dummy Activities

Case Two

- A task should be uniquely identifiable from its start node and finish node
  - *This also means that a task cannot have more than one start node and one finish node*
  - *In other words…* ***two different arrows cannot represent the same task.***

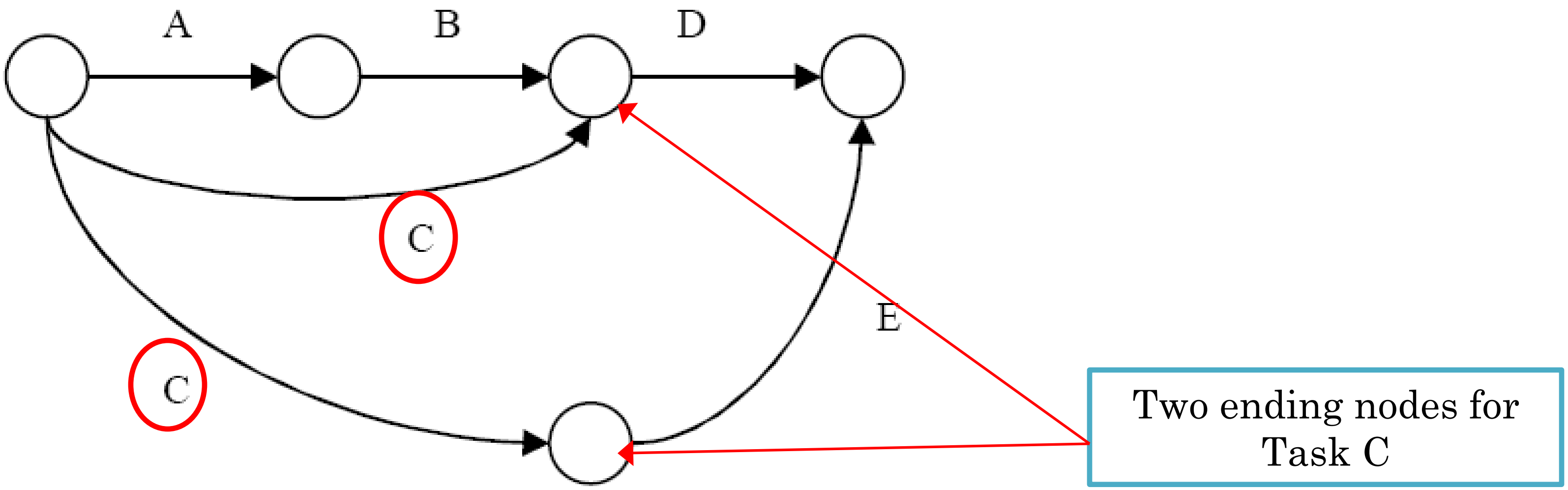
Task	Precedence
A	
B	A
C	
D	B,C
E	C



Dummy Activities

Possible Solution Two

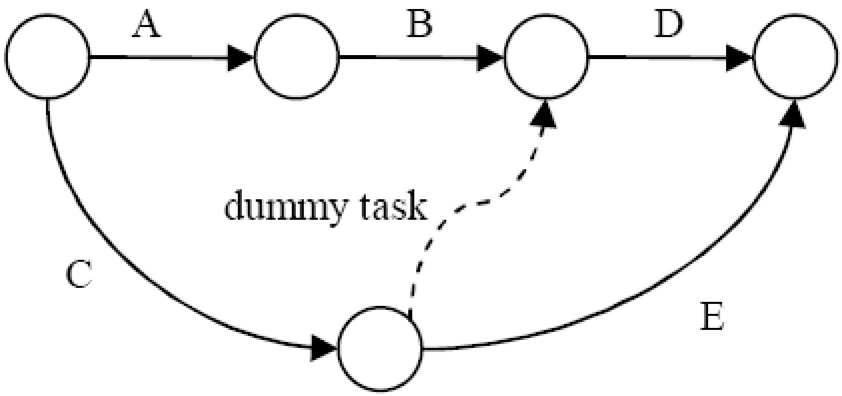
- Another option would be to insert activity E as shown below, but this provokes C to have two different end nodes.



DUMMY ACTIVITIES

Correct Solution!

- The solution is to insert a dummy task so that the precedence of E is preserved and activity C remains uniquely identifiable.

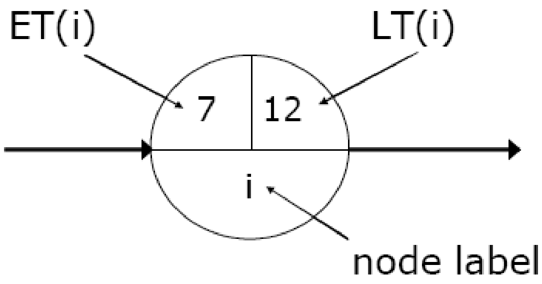


CPM – Critical Path Method

- It is determined by adding the times for the activities in each sequence.
- CPM determines the total calendar time required for the project.
- If activities outside the critical path speed up or slow down (within limits), the total project time does not change.
- The amount of time that a non-critical activity can be delayed without delaying the project is called slack-time.

CPM – Critical Path Method

- ET – Earliest node time for given activity duration and precedence relationships
- LT – Latest node time assuming no delays



- ES – Activity earliest start time
- LS – Activity latest start time
- EF – Activity earliest finishing time
- LF – Activity latest finishing time
- Slack Time – Maximum activity delay time

CPM – Critical Path Method

Step 1. Calculate ET for each node.

For each node i for which predecessors j are labelled with ET(j), ET(i) is given
by:

$$ET(i) = \max_j [ET(j) + t(j,i)]$$

where t(j,i) is the duration of task between nodes (j,i).

Step 2. Calculate LT for each node.

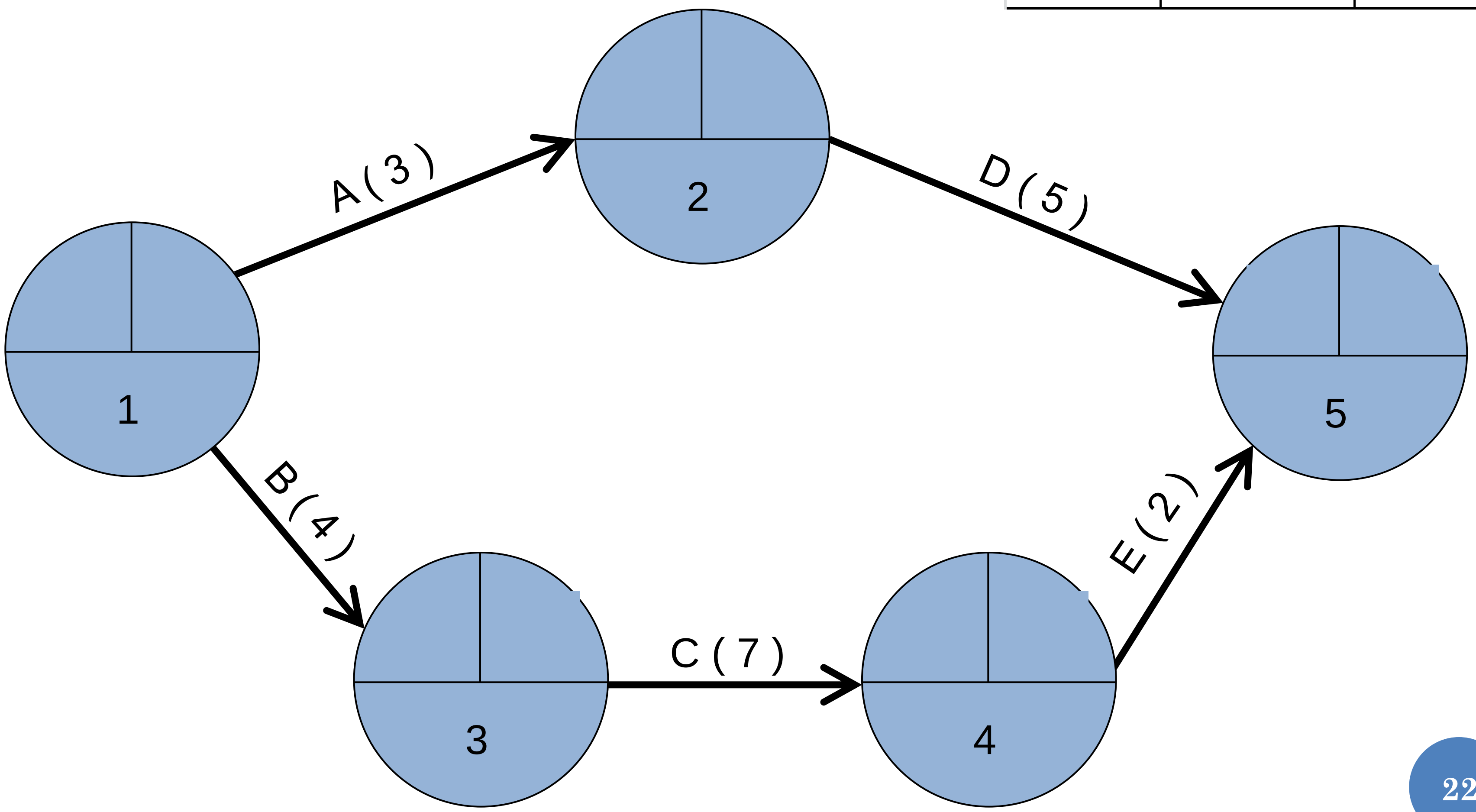
For each node i for which successors j are labelled with LT(j), LT(i) is given by:

$$LT(i) = \min_j [LT(j) – t(i,j)]$$

where t(j,i) is the duration of task between nodes (i,j).

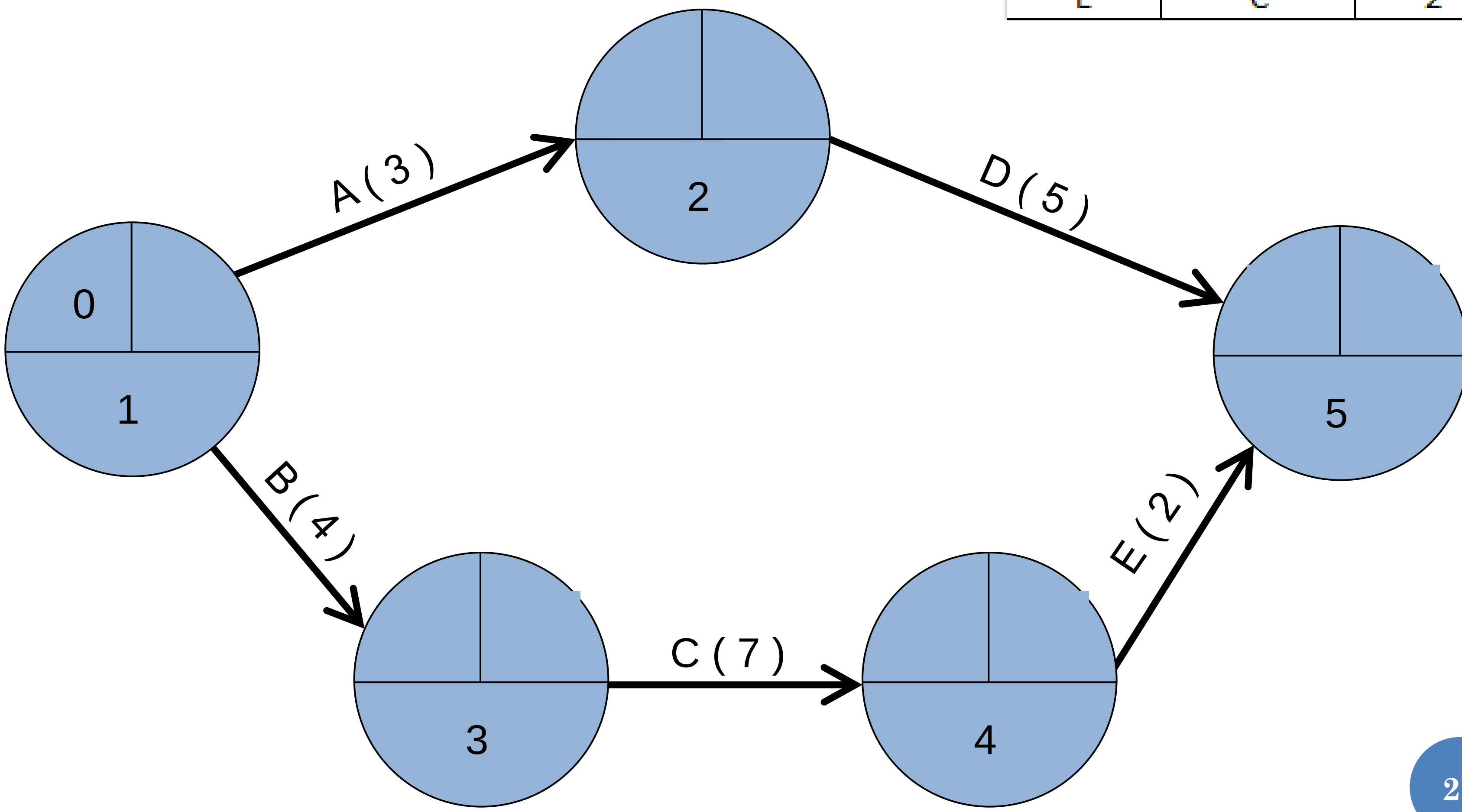
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



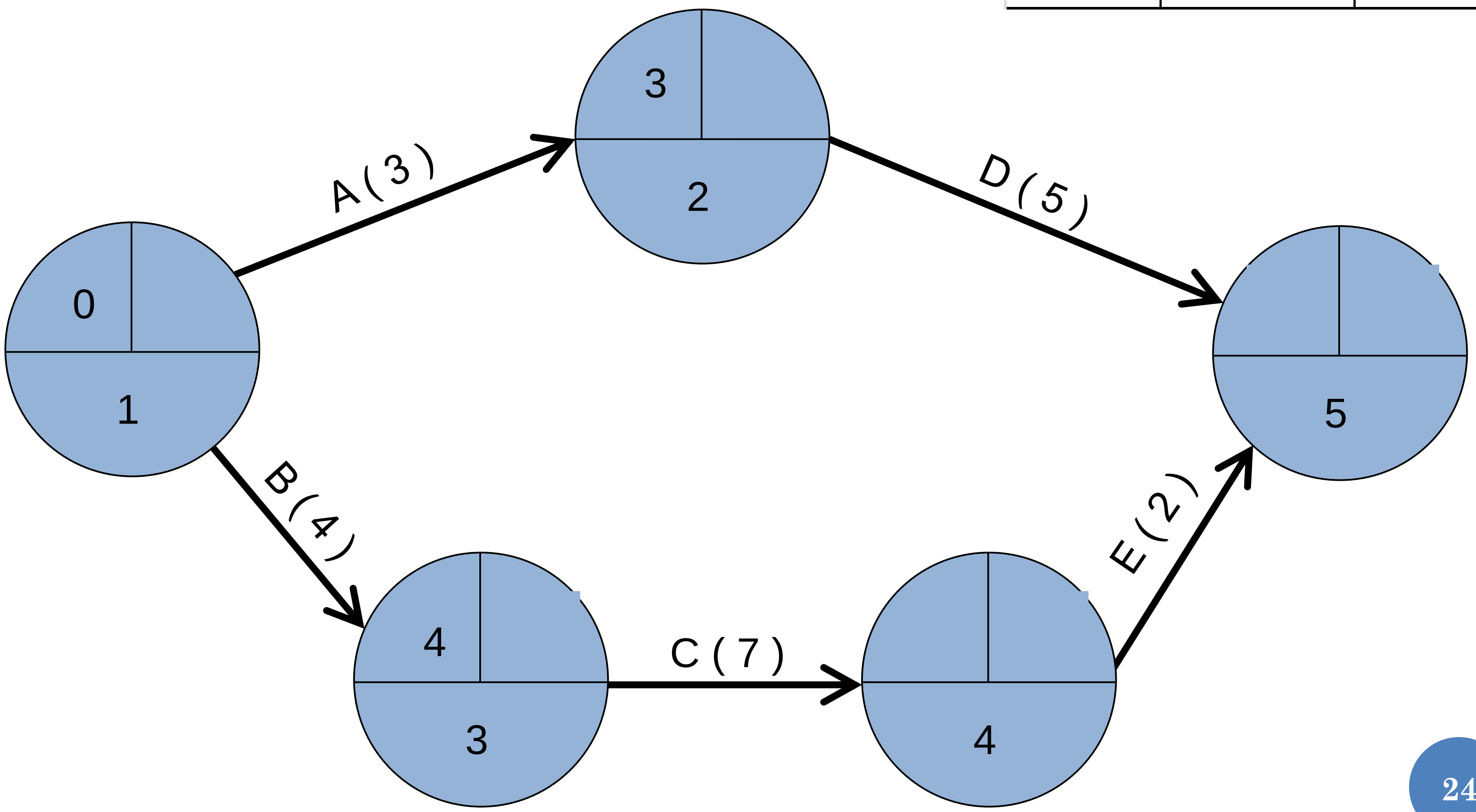
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



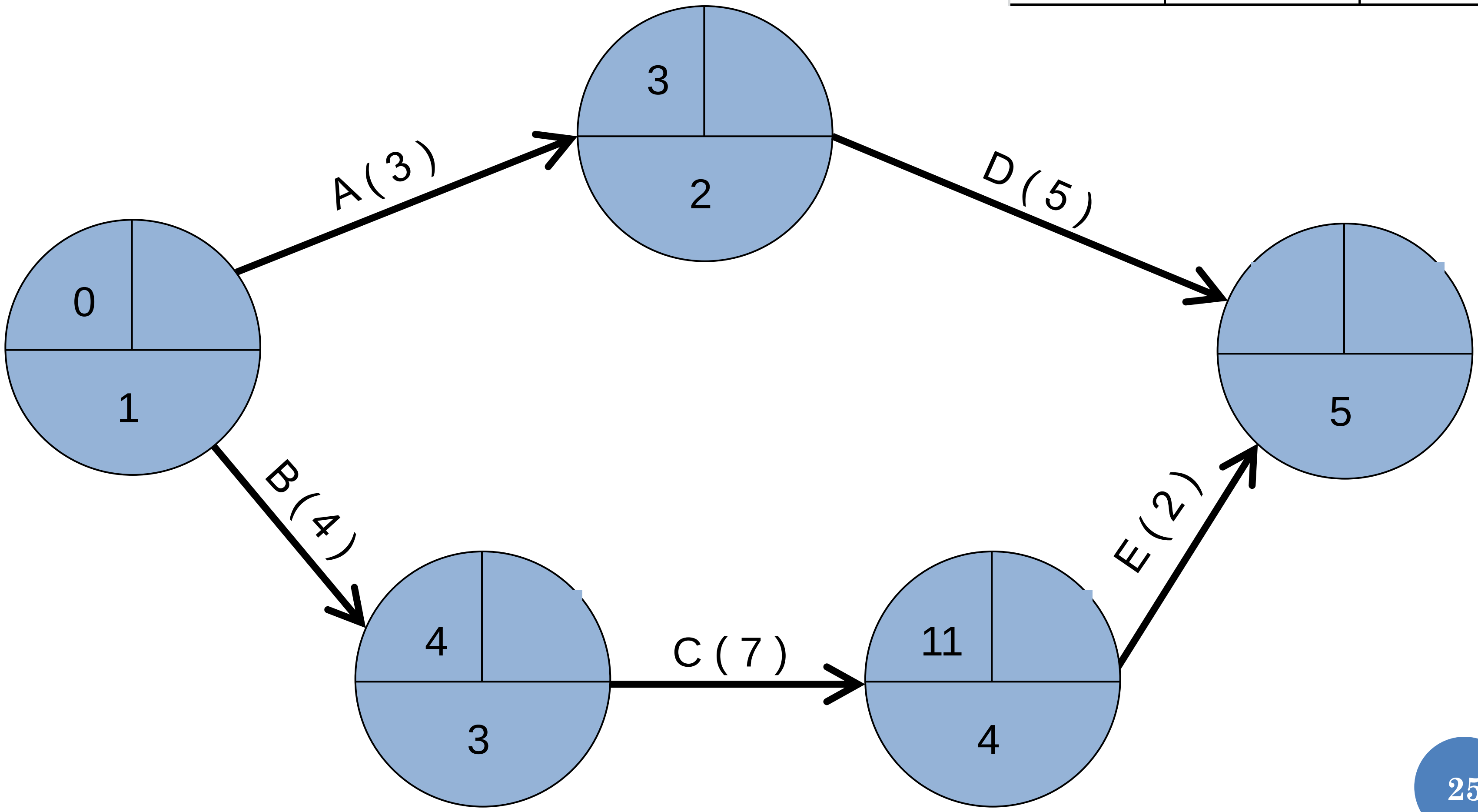
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



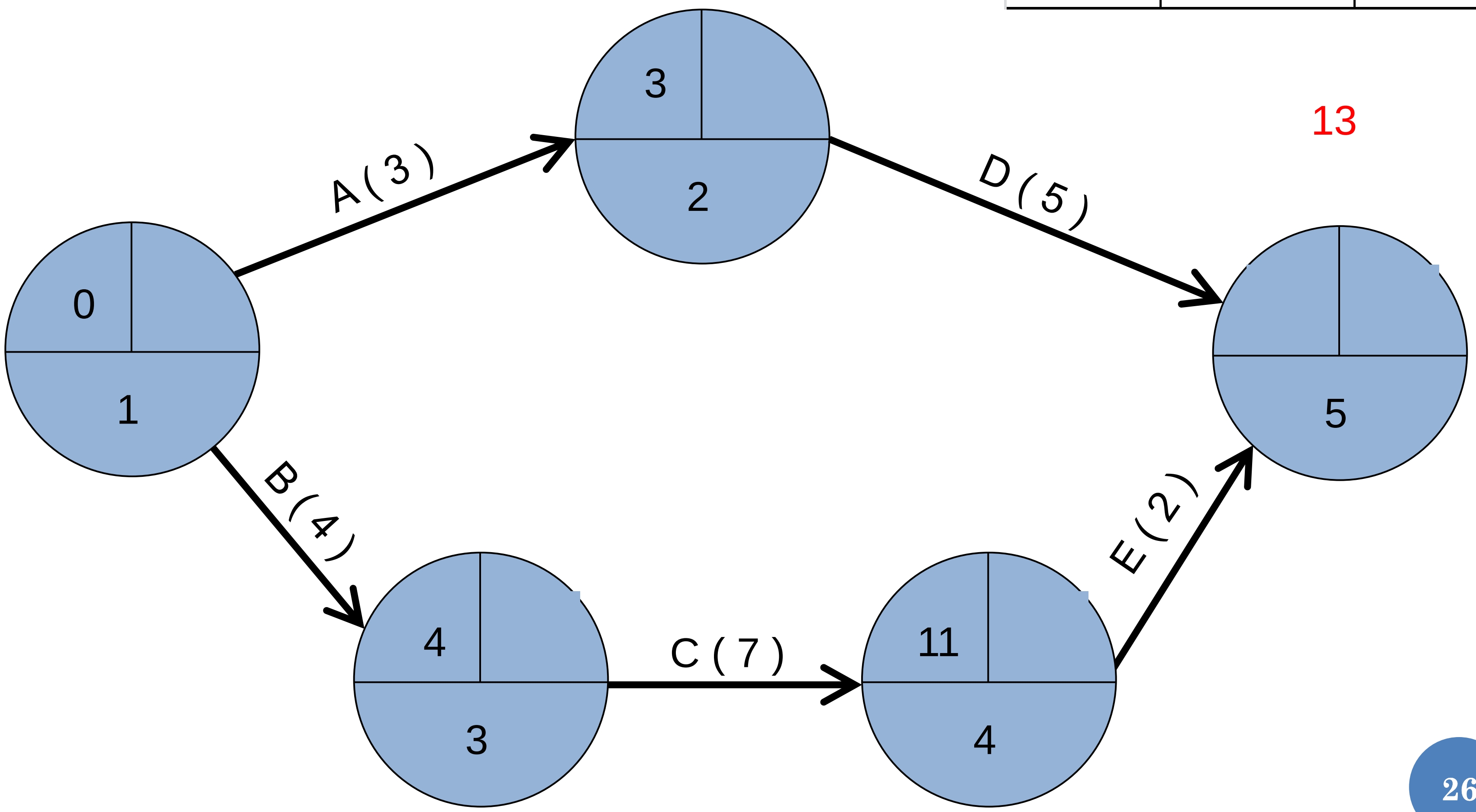
CPM – CRITICAL PATH METHOD

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



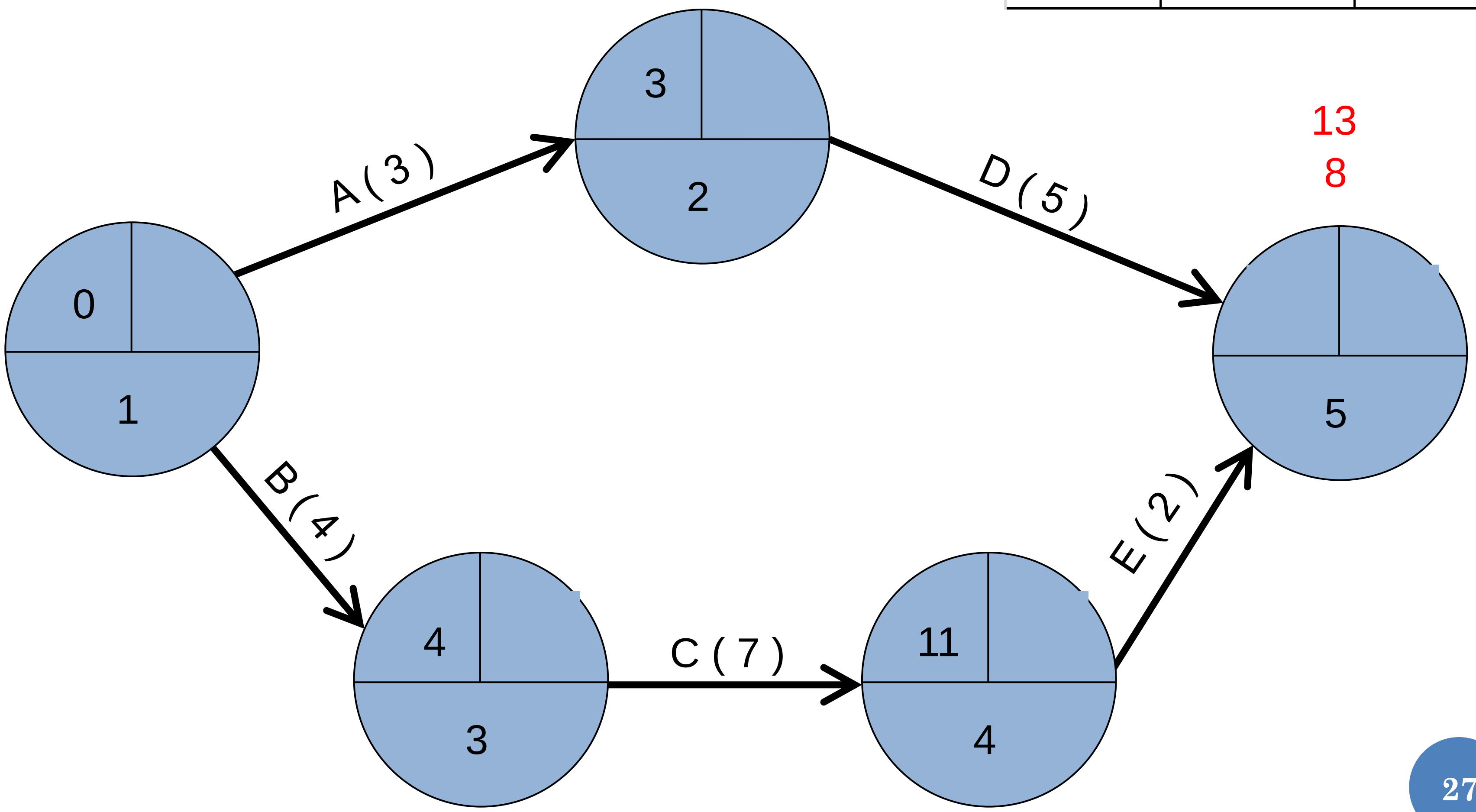
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



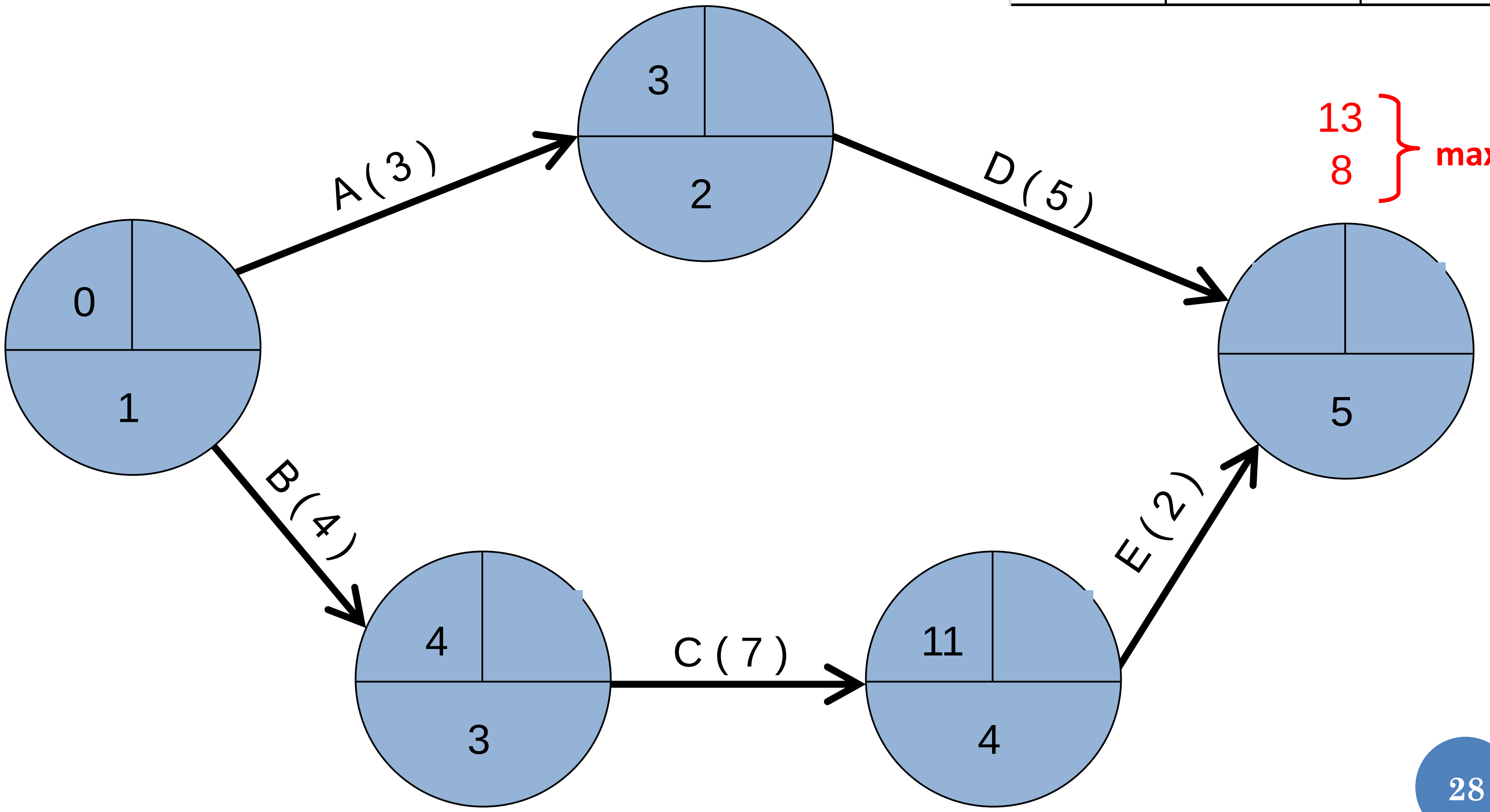
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



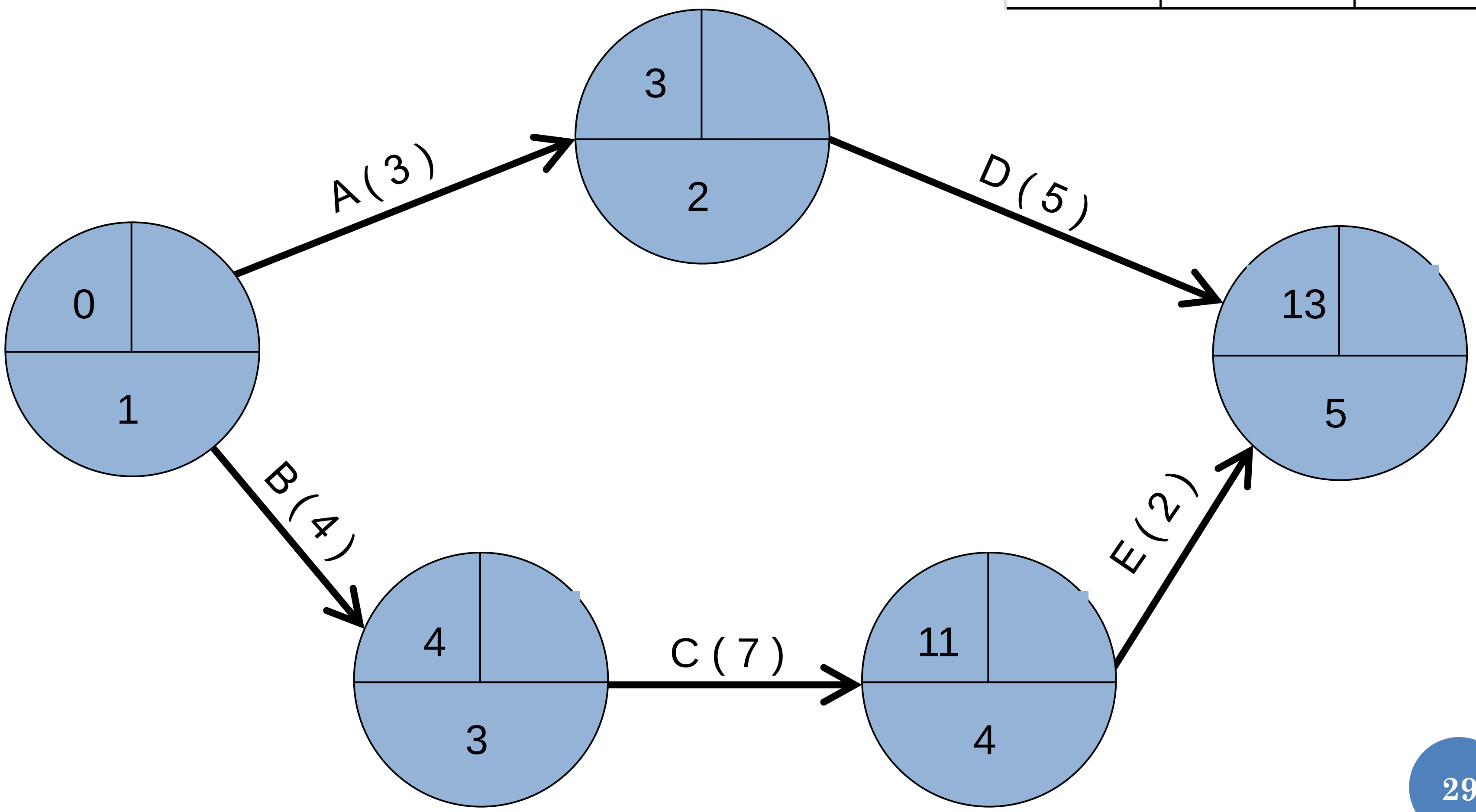
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



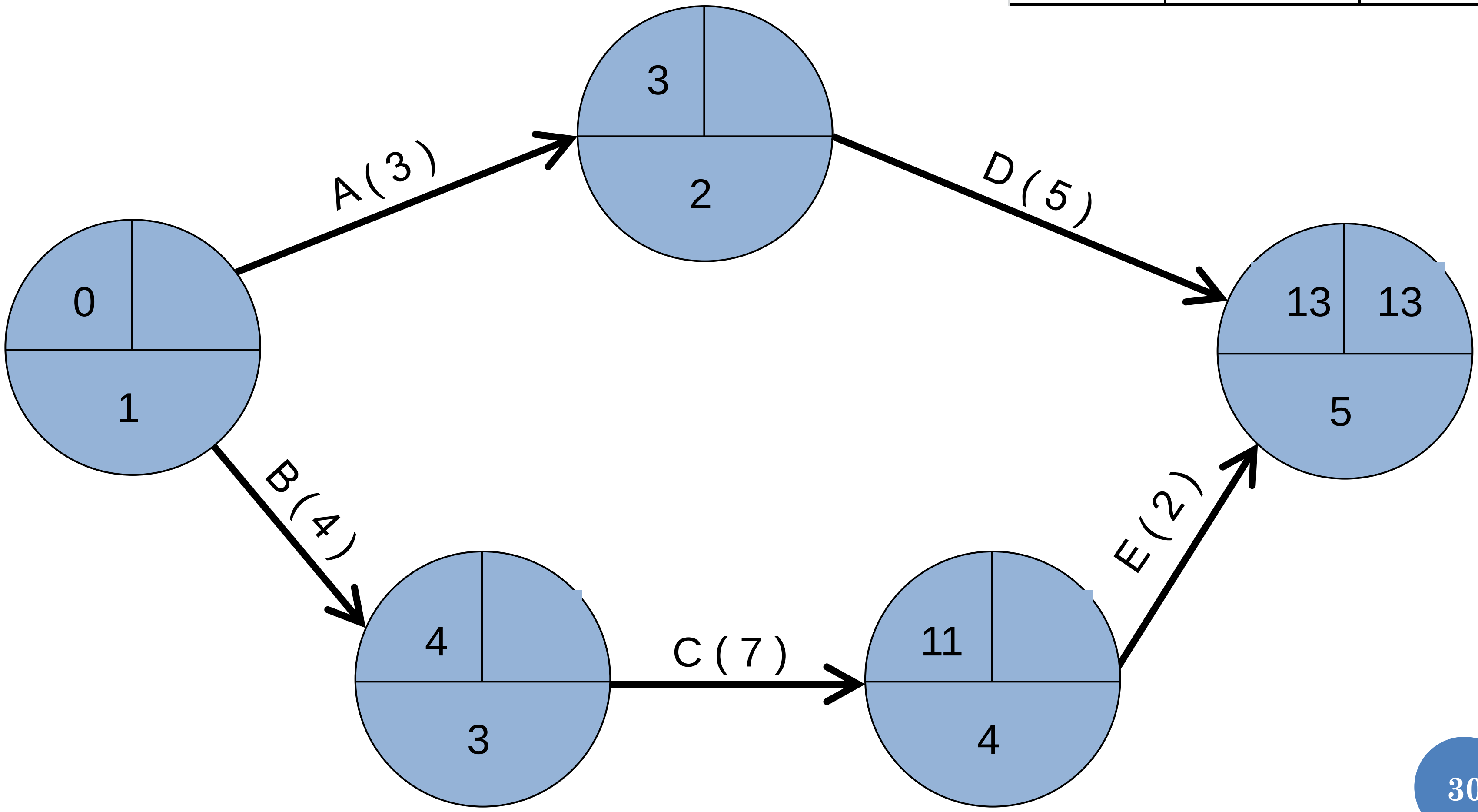
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



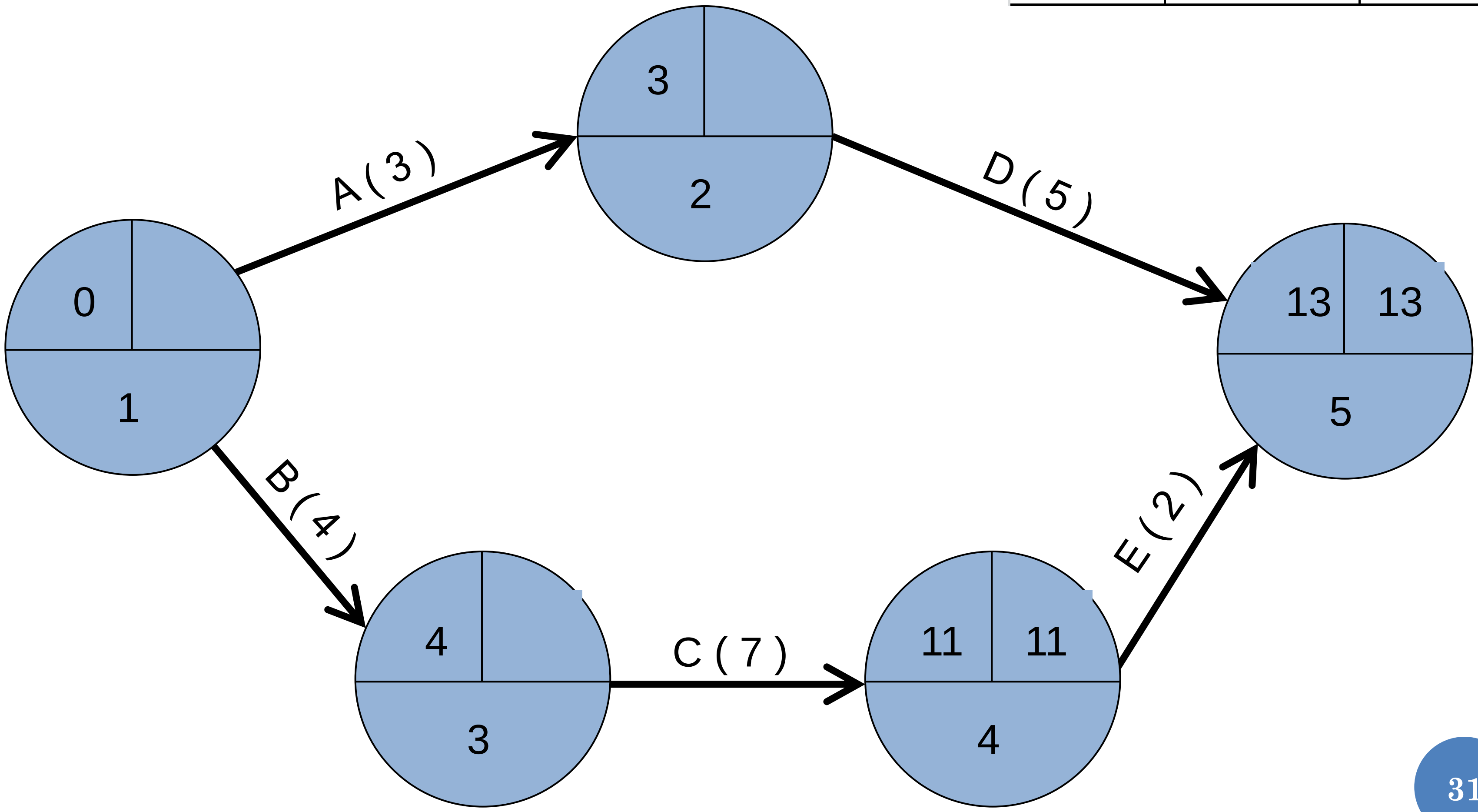
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



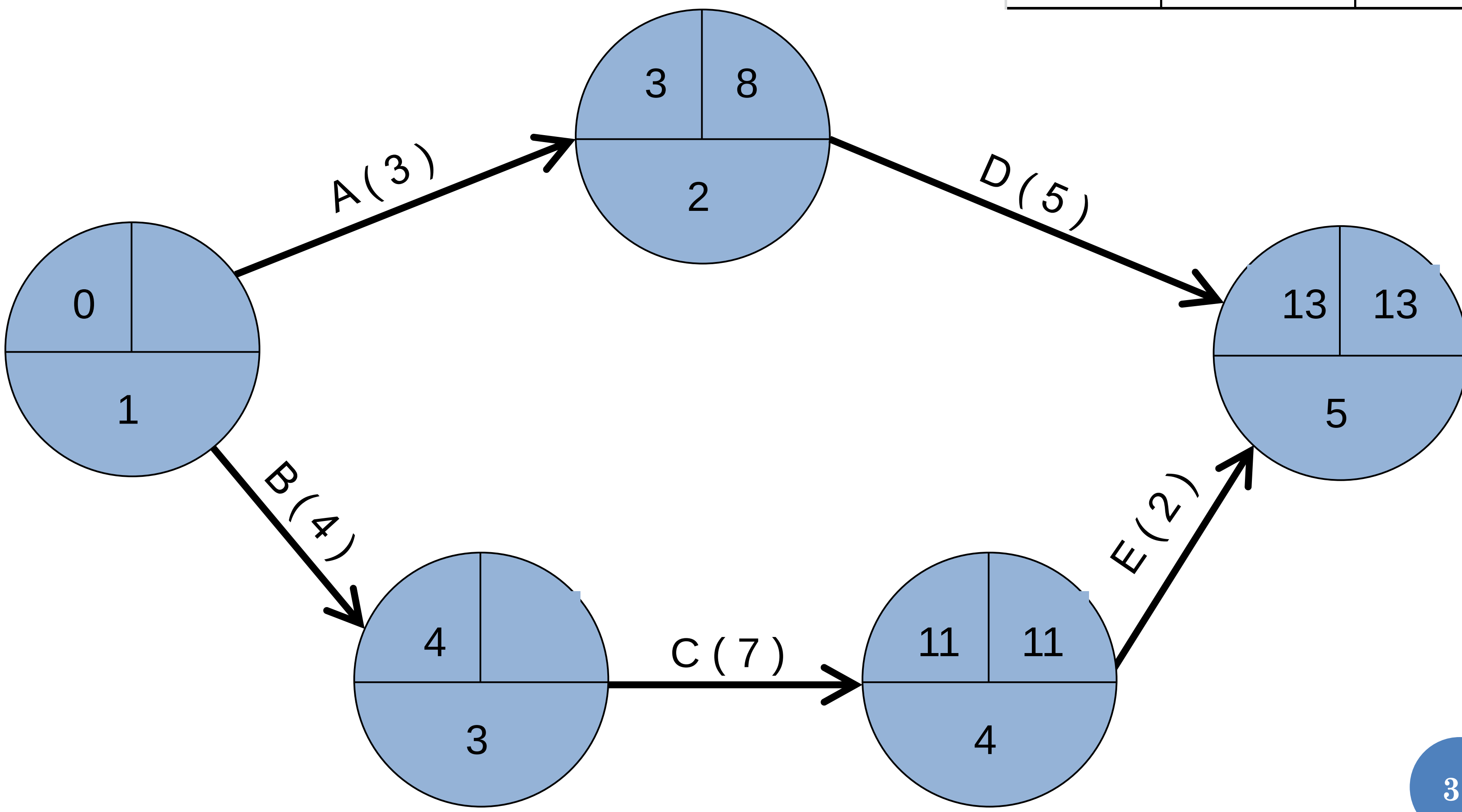
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



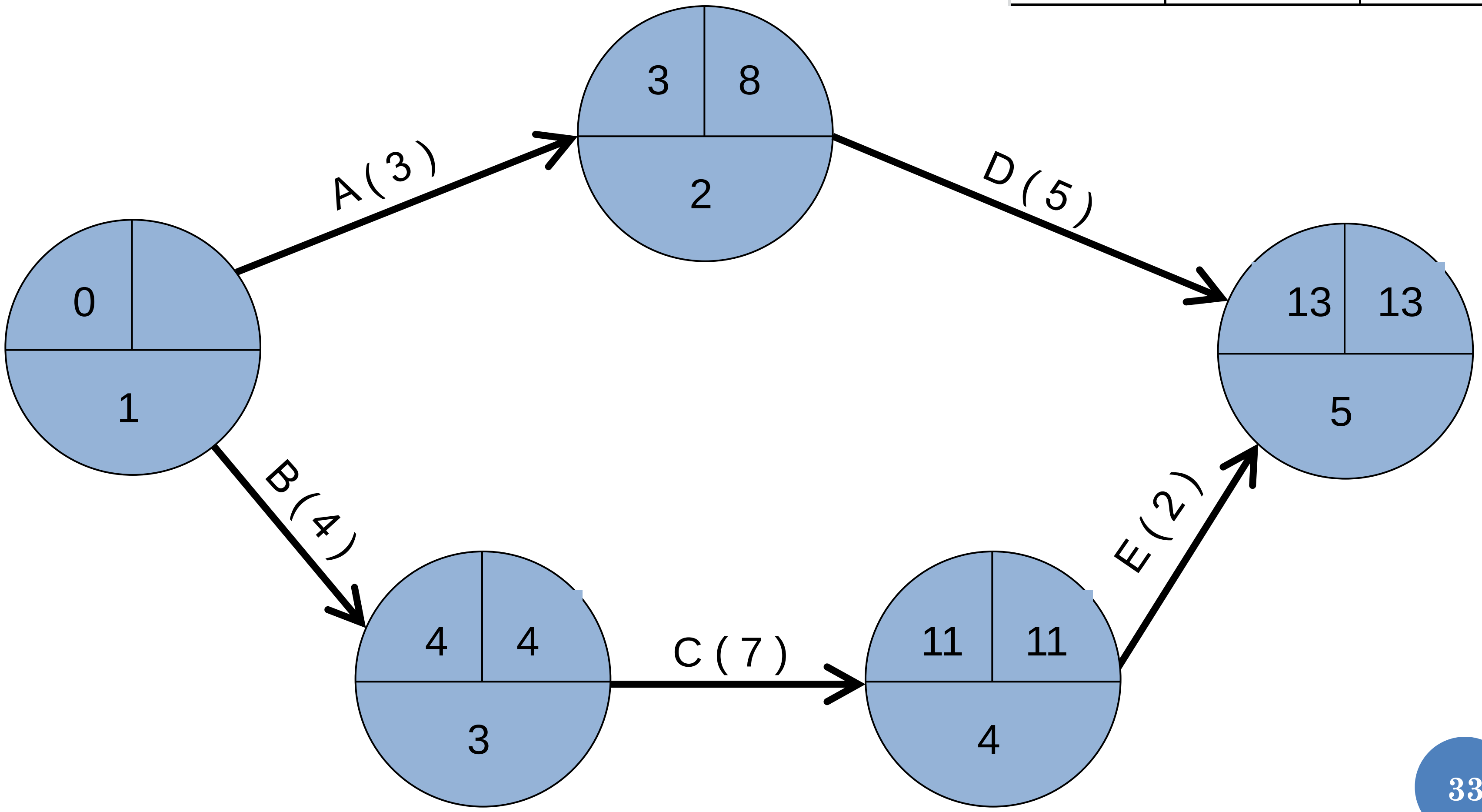
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



CPM – Critical Path Method

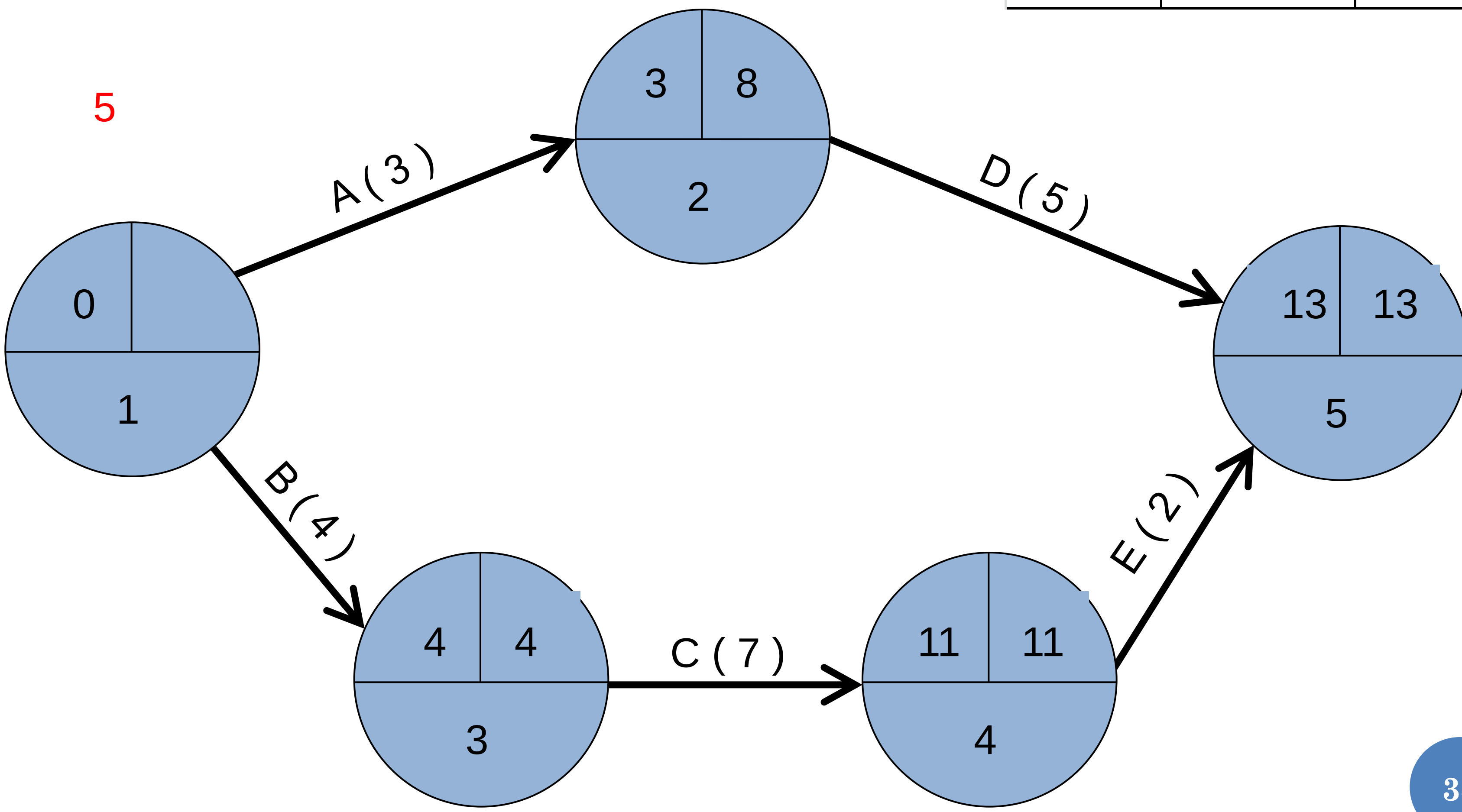
Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



CPM – Critical Path Method

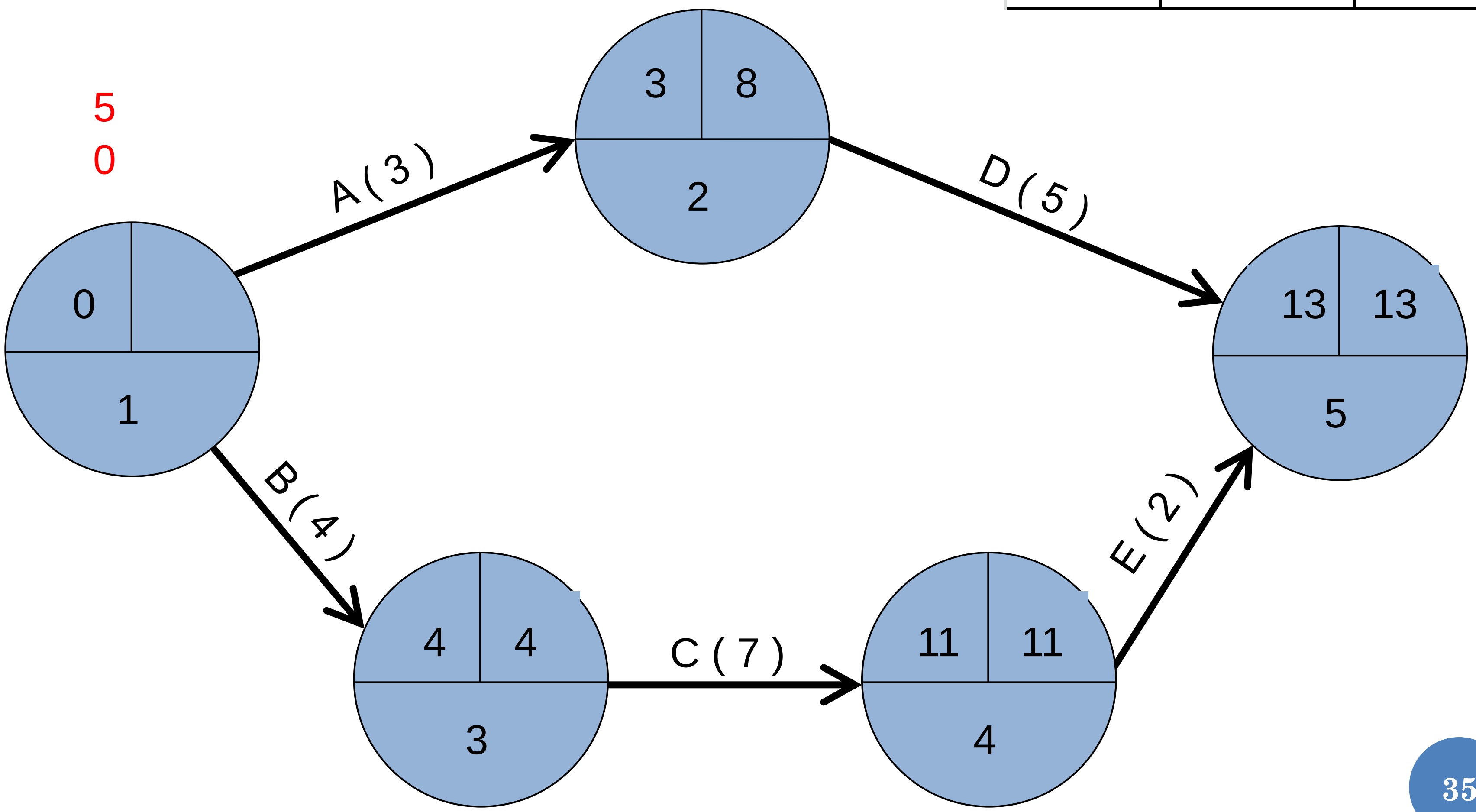
Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2

5



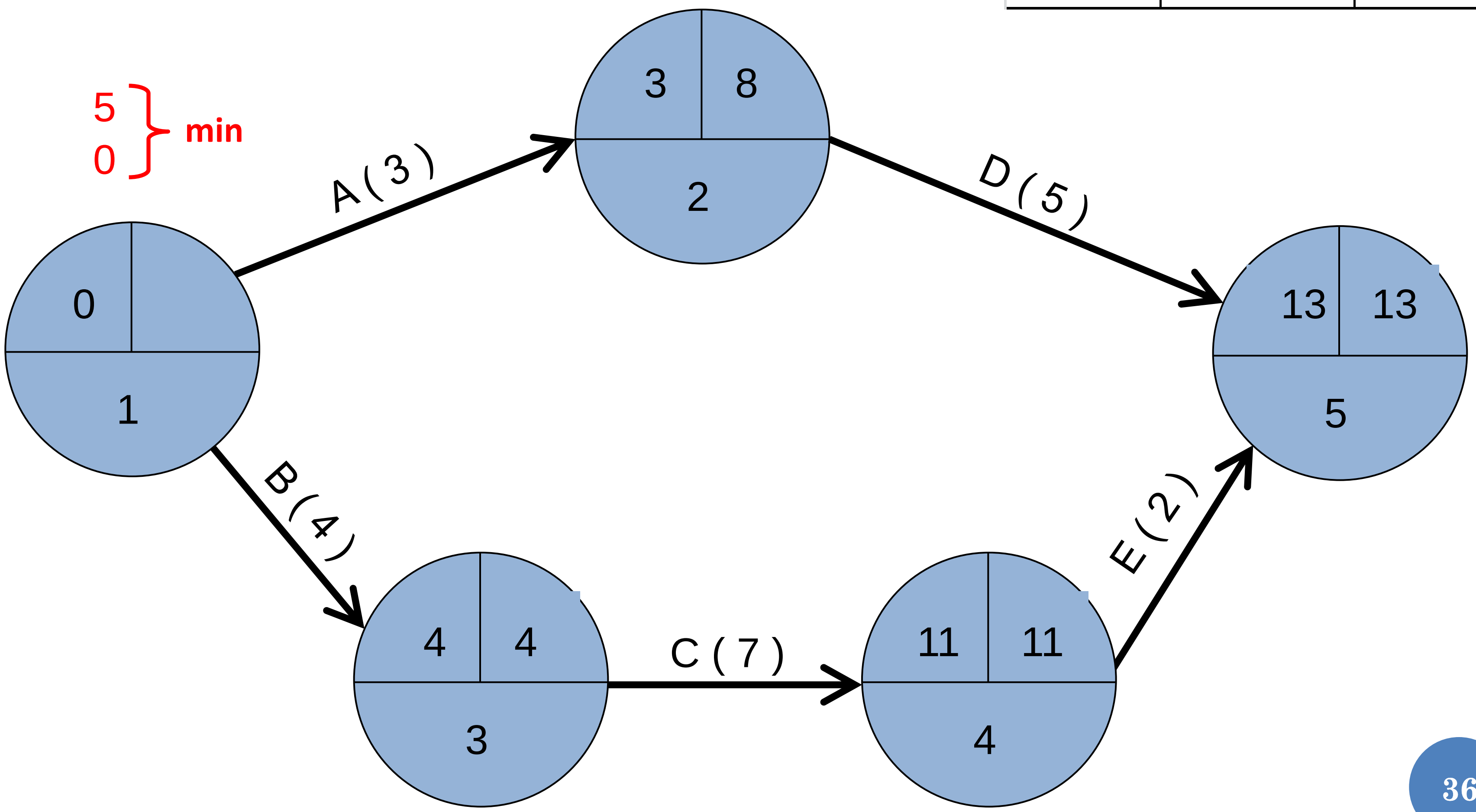
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



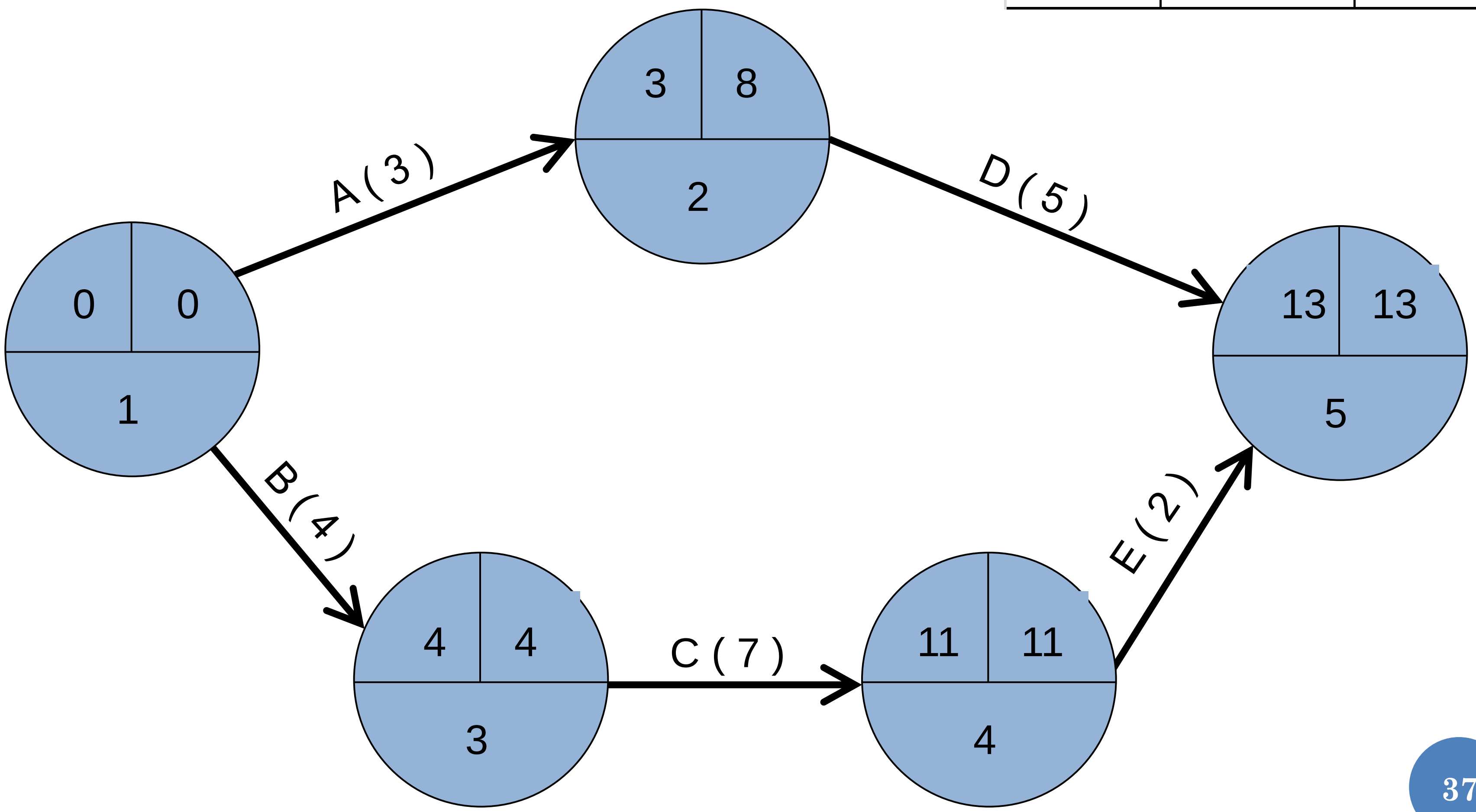
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



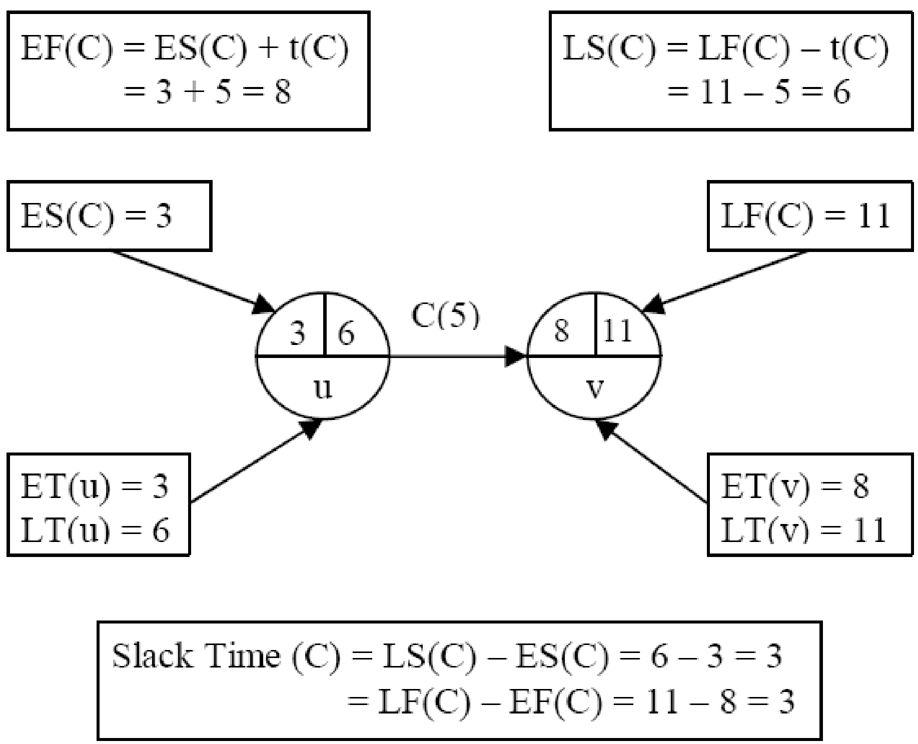
CPM – Critical Path Method

Task	Precedence	Duration
A	-	3
B	-	4
C	B	7
D	A	5
E	C	2



CPM – Critical Path Method

- An activity with zero slack time is a critical activity and *cannot be delayed* without causing a delay in the whole project.



CPM – Critical Path Method

Step 3. Calculate processing times for each activity.

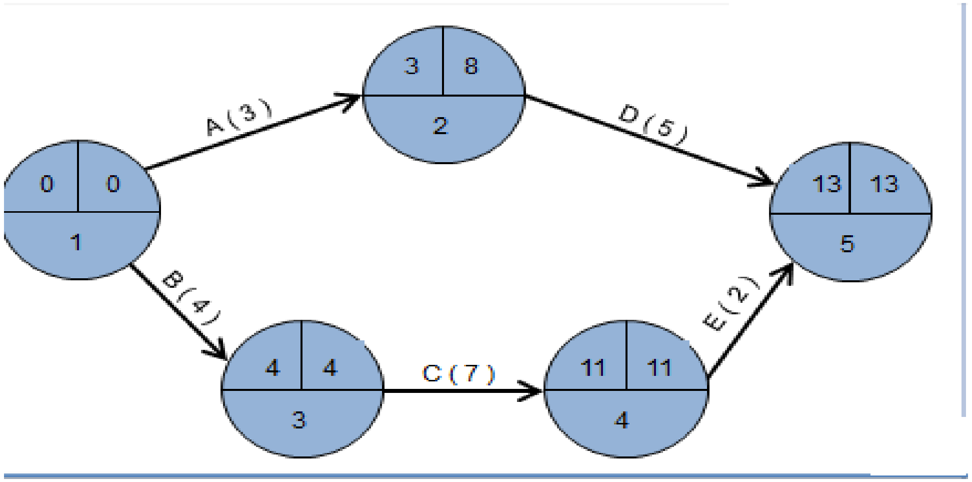
For each activity X with start node i and end node j:

$$ES(X) = ET(i)$$
$$EF(X) = ES(X) + t(X)$$
$$LF(X) = LT(j)$$
$$LS(X) = LF(X) - t(X)$$
$$\text{Slack Time } (X) = LS(X) - ES(X) = LF(X) - EF(X)$$

Where t(X) is the duration of activity X.

An activity with zero slack time is a critical activity and cannot be delayed without causing a delay in the whole project.

CPM – Critical Path Method

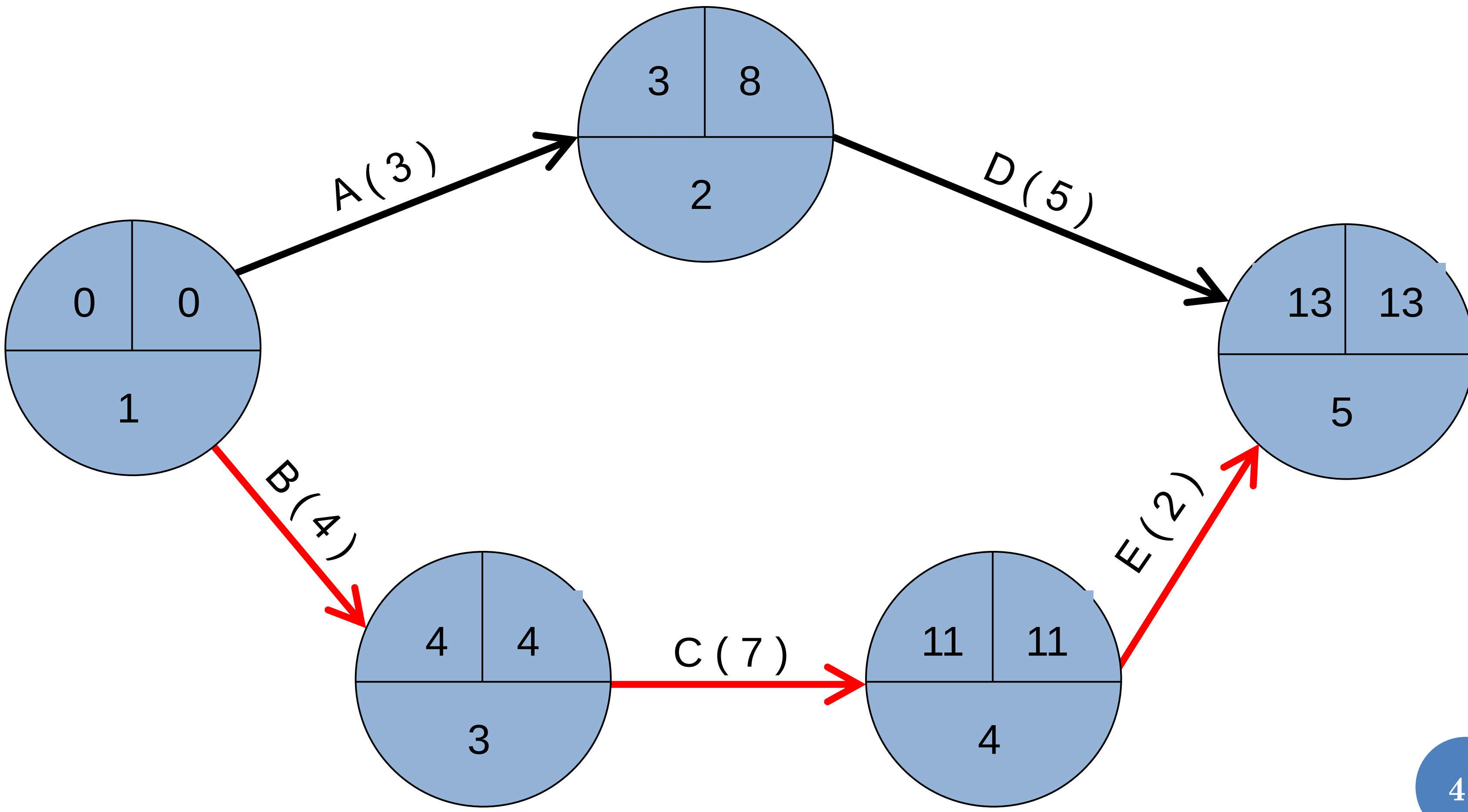


Step 3. Calculate processing times for each activity.

Task	Duration	ES	EF	LS	LF	Slack	Critical Task
A	3	0	3	5	8	5	No
B	4	0	4	0	4	0	Yes
C	7	4	11	4	11	0	Yes
D	5	3	8	8	13	5	No
E	2	11	13	11	13	0	Yes

Reading: (Kendall&Kendall, chapter 3), (Dennis &Wixom, chapter 3)

CPM – Critical Path Method



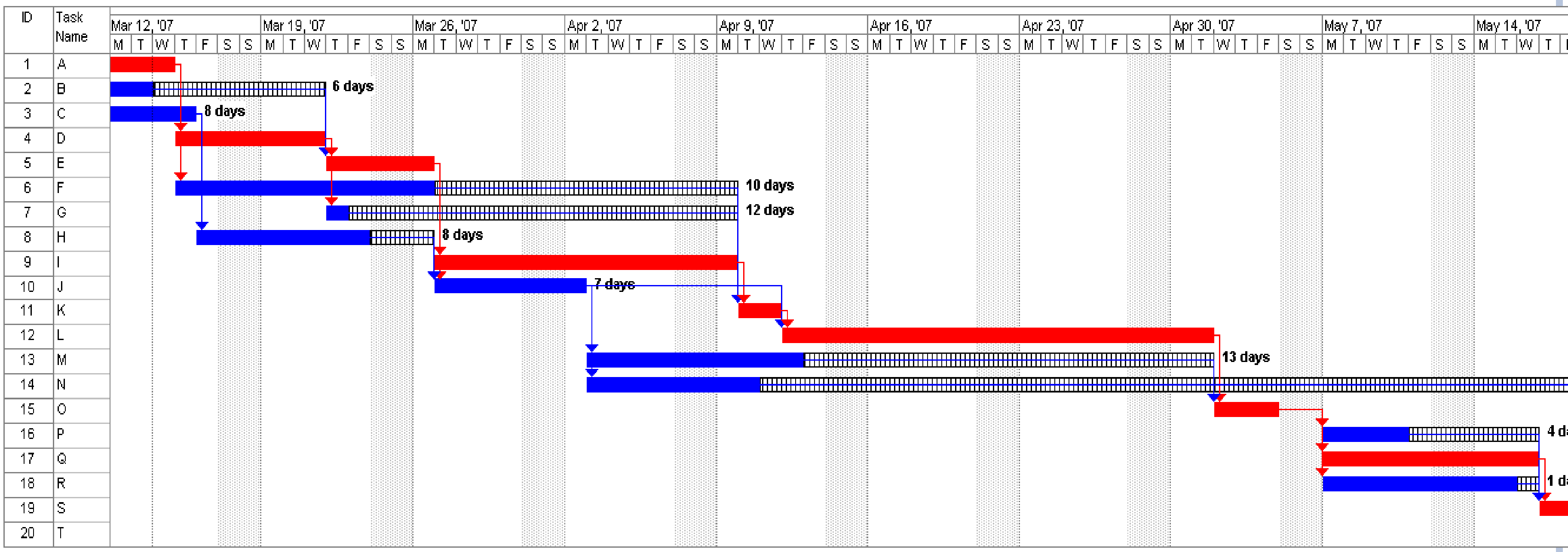
LECTURE – GANTT CHART

- GANTT Charts
- Constructing GANTT Charts
- Staffing and Re-scheduling
- Project Management Software

GANTT CHART

- A GANTT chart is a type of bar chart that illustrates a project schedule.
- After the PERT/CPM analysis is completed, the following phase is to construct the GANTT chart and then to re-allocate resources and re-schedule if necessary.
- GANTT charts have become a common technique for representing the phases and activities of a project work breakdown structure.
- It was introduced by Henry Gantt around 1910 – 1915.

GANTT CHART



ID
Task Name
Mar 12, '07
Mar 19, '07
Mar 26, '07
Apr 2, '07
Apr 9, '07
Apr 16, '07
Apr 23, '07
Apr 30, '07
May 7, '07
May 14, '07
M T W T F S S
1 A
2 B
3 C
4 D
5 E
6 F
7 G
8 H
9 I
10 J
11 K
12 L
13 M
14 N
15 O
16 P
17 Q
18 R
19 S
20 T
6 days
8 days
10 days
12 days
8 days
7 days
13 days
4 day
1 day

GANTT CHART

- Characteristics:
  - The bar in each row identifies the corresponding task
  - The horizontal position of the bar identifies start and end times of the task
  - Bar length represents the duration of the task
  - Task durations can be compared easily
  - Good for allocating resources and re-scheduling
  - Precedence relationships can be represented using arrows
  - Critical activities are usually highlighted
  - Slack times are represented using bars with doted lines
  - The bar of each activity begins at the activity earliest start time (ES)
  - The bar of each activity ends at the activity latest finish time (LF).

Gantt Chart

- Advantages
  - Simple
  - Good visual communication to others
  - Task durations can be compared easily
  - Good for scheduling resources

- Disadvantages
  - Dependencies are more difficult to visualise
  - Minor changes in data can cause major changes in the chart

Constructing Gantt Chart

- The steps to construct a GANTT chart from the information obtained by PERT/CPM are:
  1. Schedule the critical tasks in the correct position.
  2. Place the time windows in which the non-critical tasks can be scheduled.
  3. Schedule the non-critical tasks according to their earliest starting times.
  4. Indicate precedence relationships between tasks.

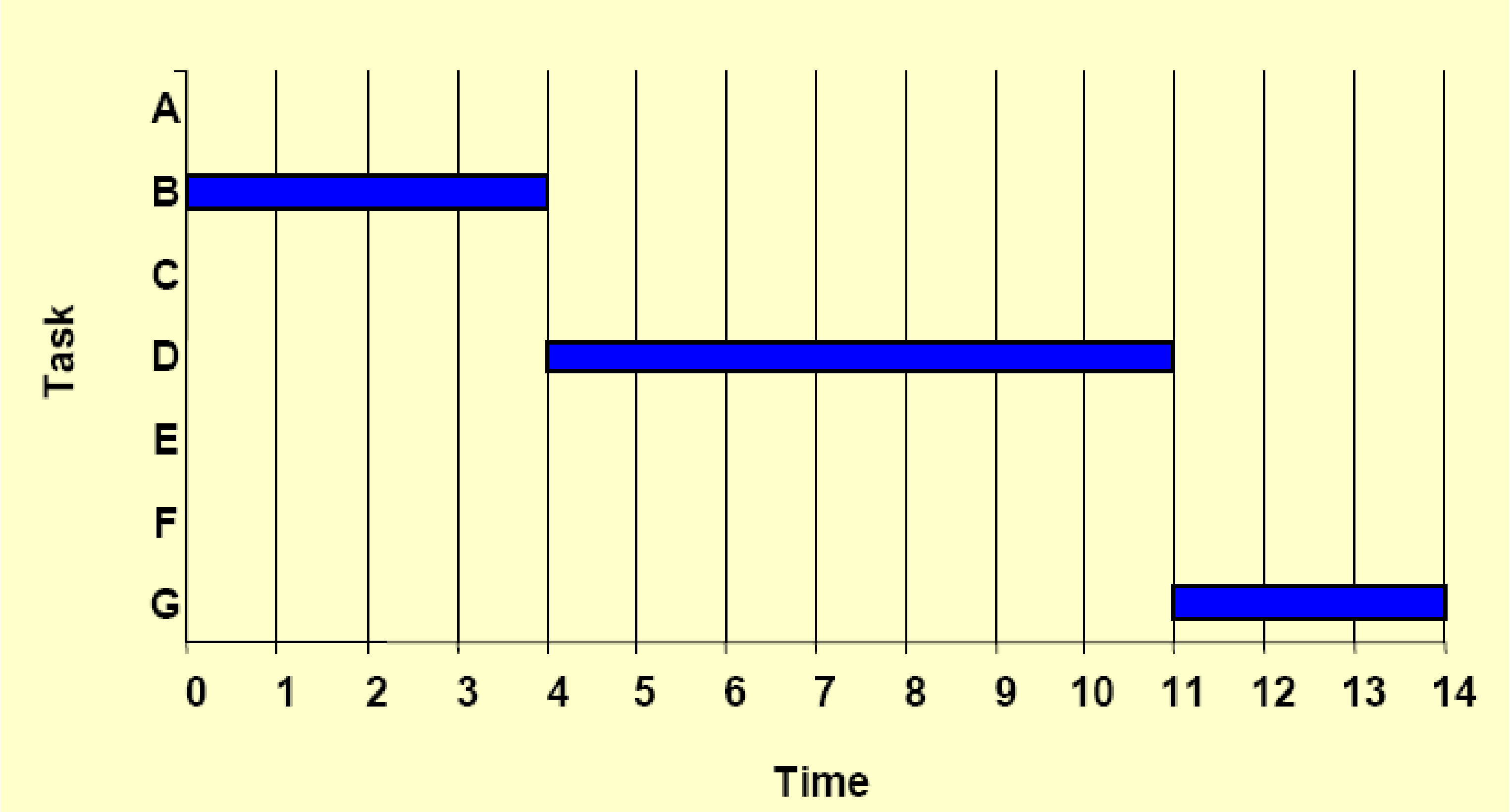
CONSTRUCTING GANTT CHART

- Example of an early GANTT chart construction:

Task	Duration	Precedence	ES	EF	LS	LF	Slack Time	Critical Task
A	3		0	3	3	6	3	N
B	4		0	4	0	4	0	Y
C	5	A	3	8	6	11	3	N
D	7	B	4	11	4	11	0	Y
E	2	B	4	6	8	10	4	N
F	4	E	6	10	10	14	4	N
G	3	C,D	11	14	11	14	0	Y

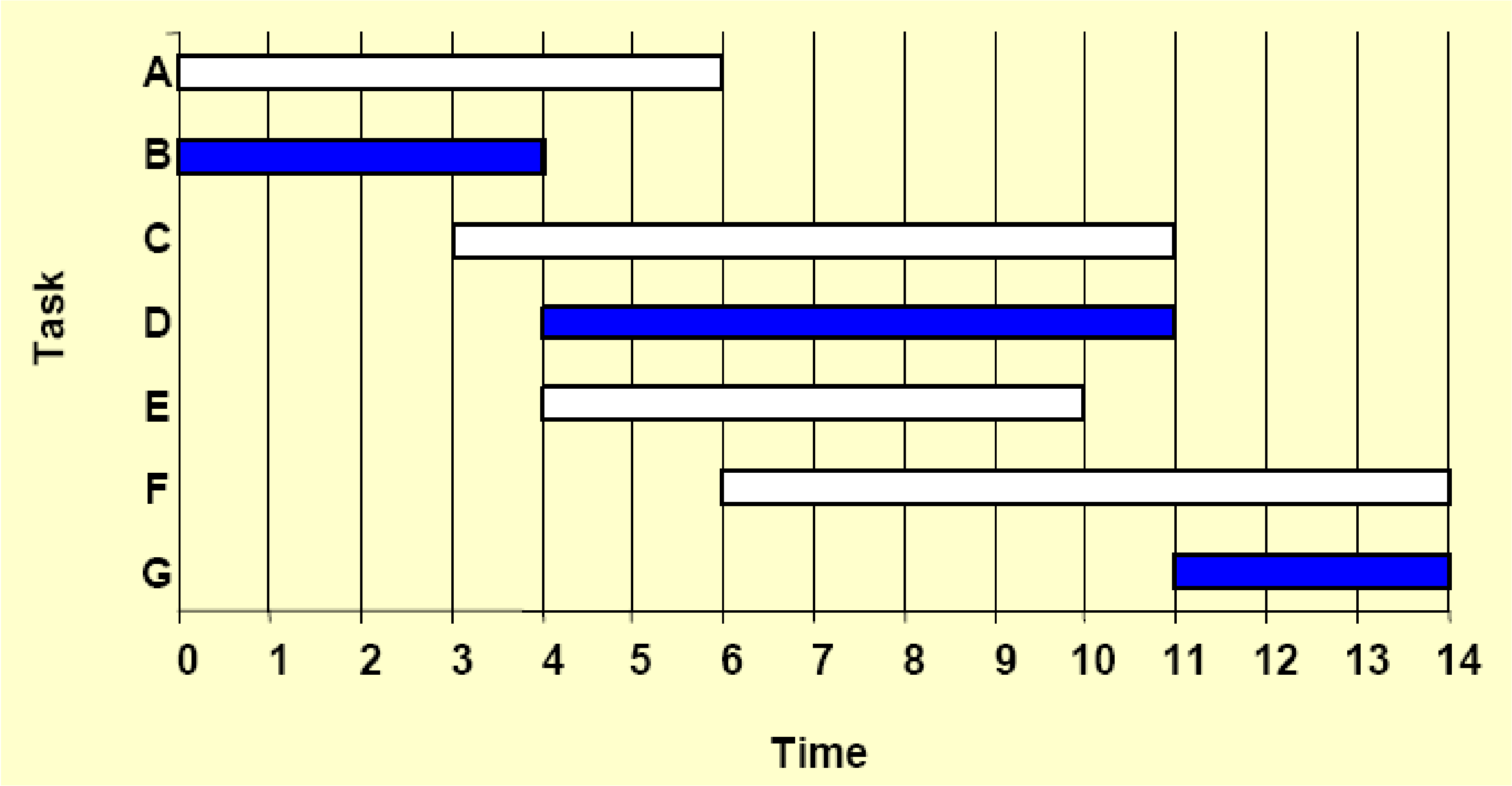
Constructing Gantt Chart

Step 1. Schedule critical tasks:



Constructing Gantt Chart

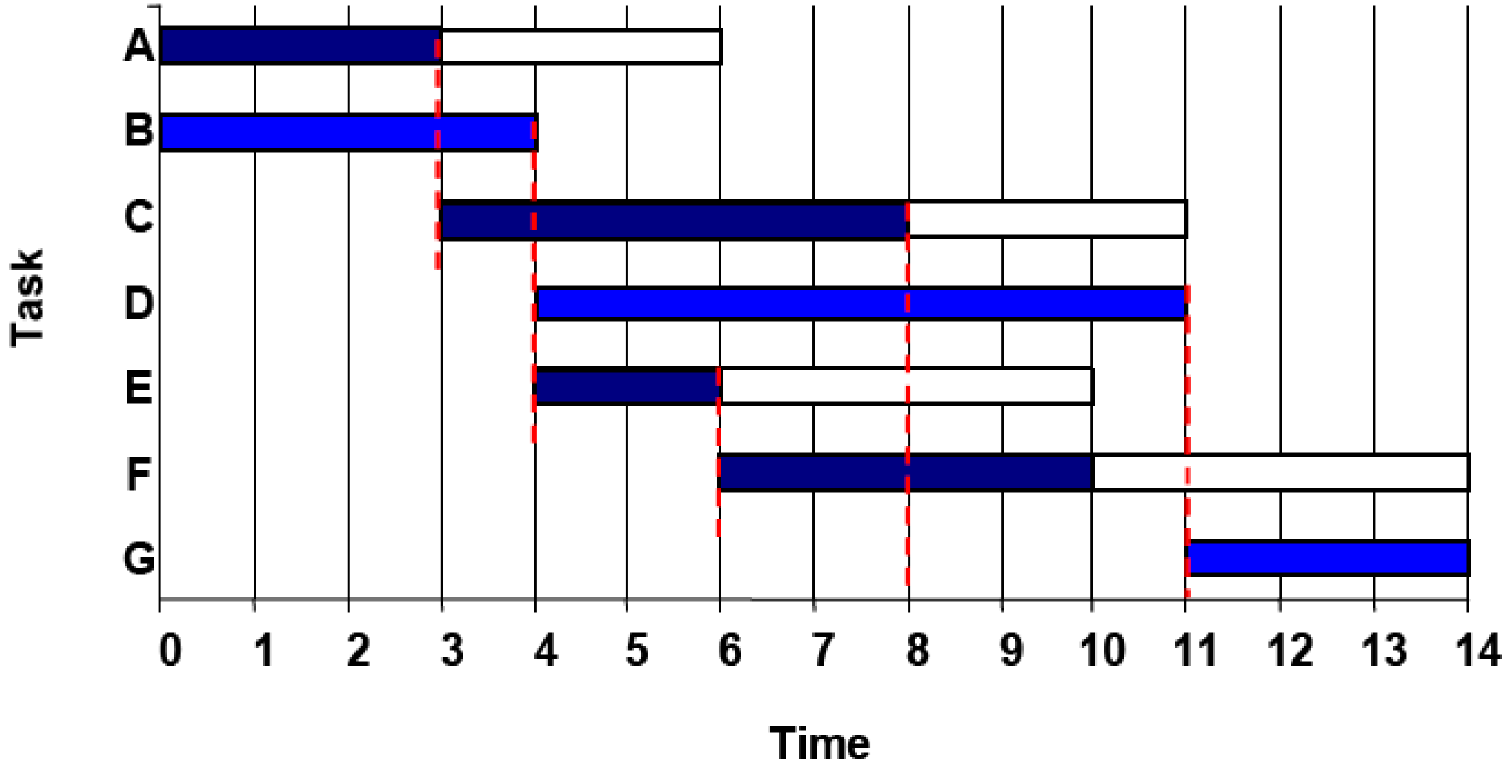
Step 2. Place time windows for non-critical tasks:



Constructing Gantt Chart

Step 3. Schedule non-critical tasks

Step 4. Indicate precedence relationships:



Staffing & Re-scheduling

- Once the project schedule, (*e.g. GANTT chart),* has been constructed, take into account
  - available staff hours
  - slack times and
  - the project schedule

Assign staff and other resources to each activity in the project

STAFFING & RE-SCHEDULING

- Resource Smoothing is a technique used to *re-allocate resources* and *re-schedule activities.*
- In resource smoothing, non-critical tasks are re-scheduled within their time window.
- Staff Utilization*:(duration of activity x staff required for each activity*, all added together) / (*maximum staff required x duration of project*)

Staffing & Re-scheduling

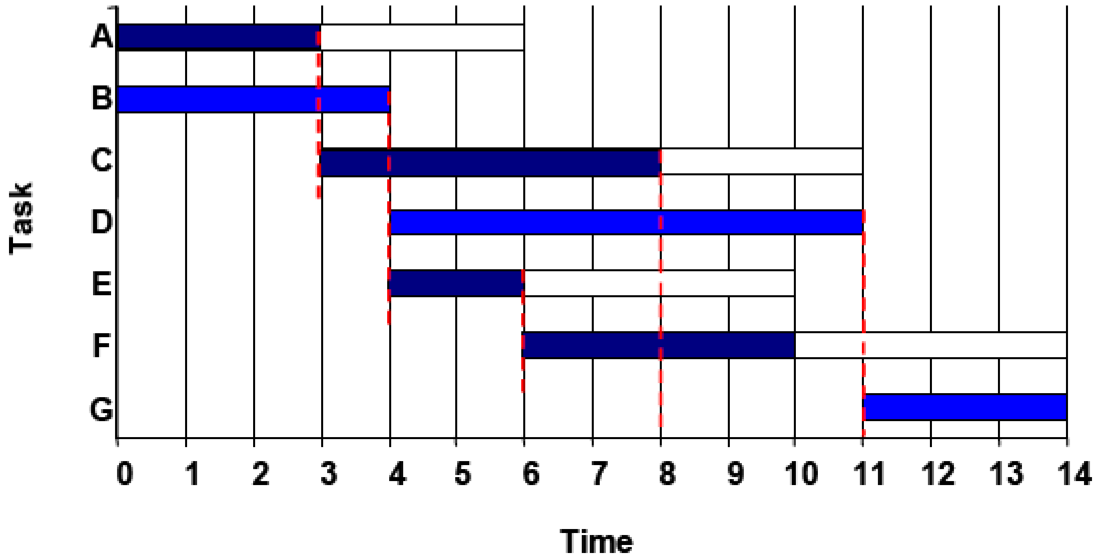
Example1

Task	Duration	Precedence	ES	EF	LS	LF	Slack Time	Critical Task
A	3		0	3	3	6	3	N
B	4		0	4	0	4	0	Y
C	5	A	3	8	6	11	3	N
D	7	B	4	11	4	11	0	Y
E	2	B	4	6	8	10	4	N
F	4	E	6	10	10	14	4	N
G	3	C,D	11	14	11	14	0	Y

STAFFING & RE-SCHEDULING

The original schedule (constructed above) for this project is as shown below.

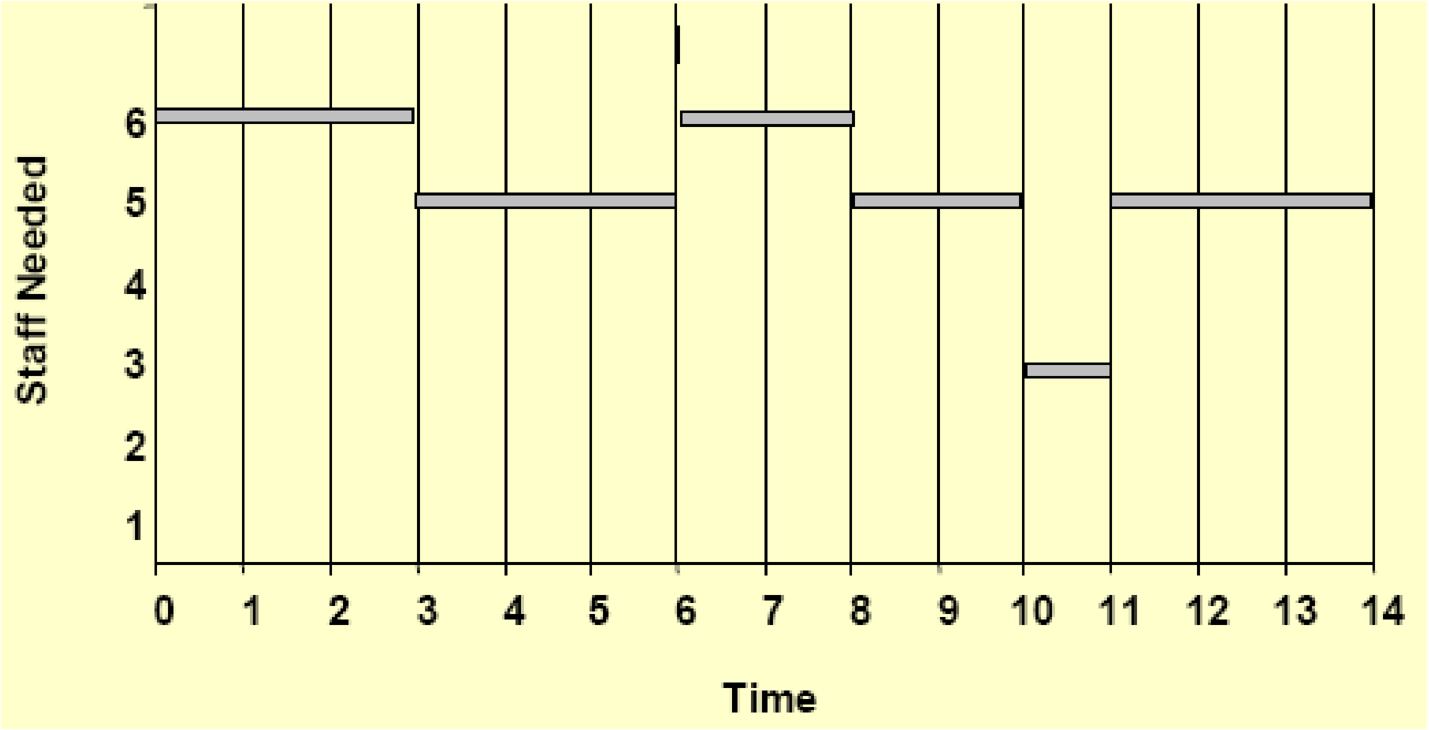
Task	Duration	Staff Needed
A	3	2
B	4	4
C	5	1
D	7	3
E	2	1
F	4	2
G	3	5



1. Staff utilisation = (3x2+4x4+5x1+7x3+2x1+4x2+3x5)/(14x6) = 0.857 = 85.5%

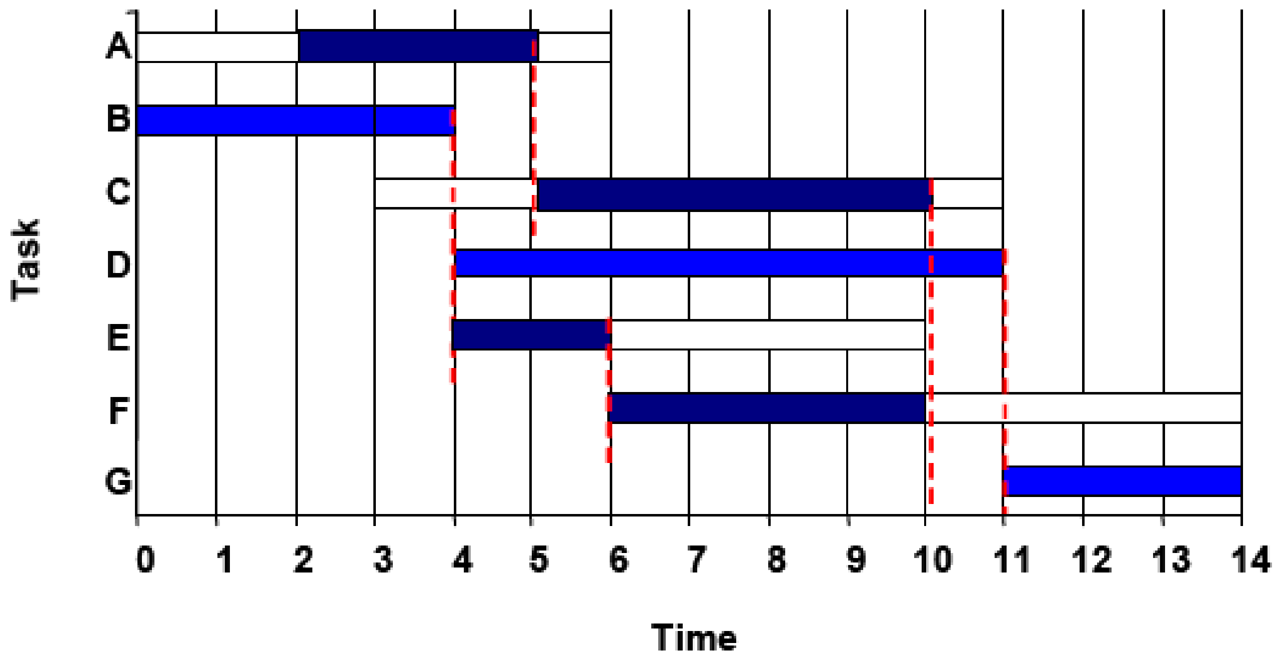
Staffing & Re-scheduling

2. Work out the Staff Profile



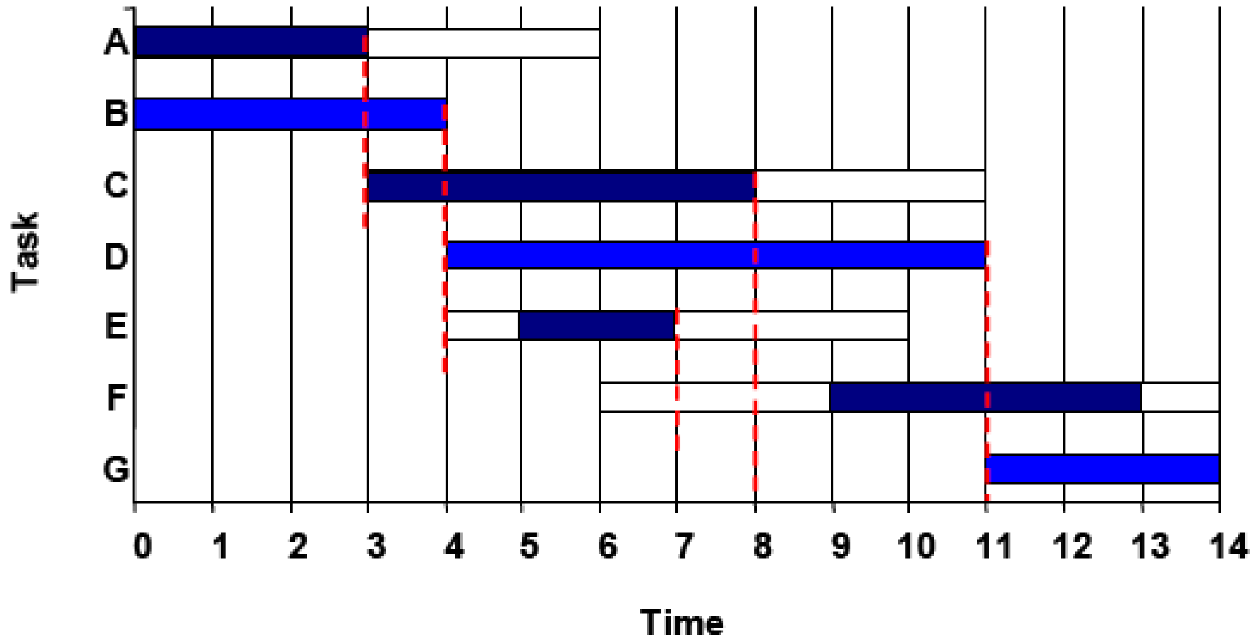
Staffing & Re-scheduling

- Now, assume that there are 6 people available for working in this project but one of them returns from holidays at time=2.
- So *re-scheduling* is needed because activities A and B cannot be carried out in parallel until time=2.



STAFFING & RE-SCHEDULING

- Suppose another scenario in which equipment and materials needed to carry out activities E and F are available at time=5 and time=9 respectively instead of being available at the activities ES time. Then, rescheduling is needed but the overall duration of the project is not affected.



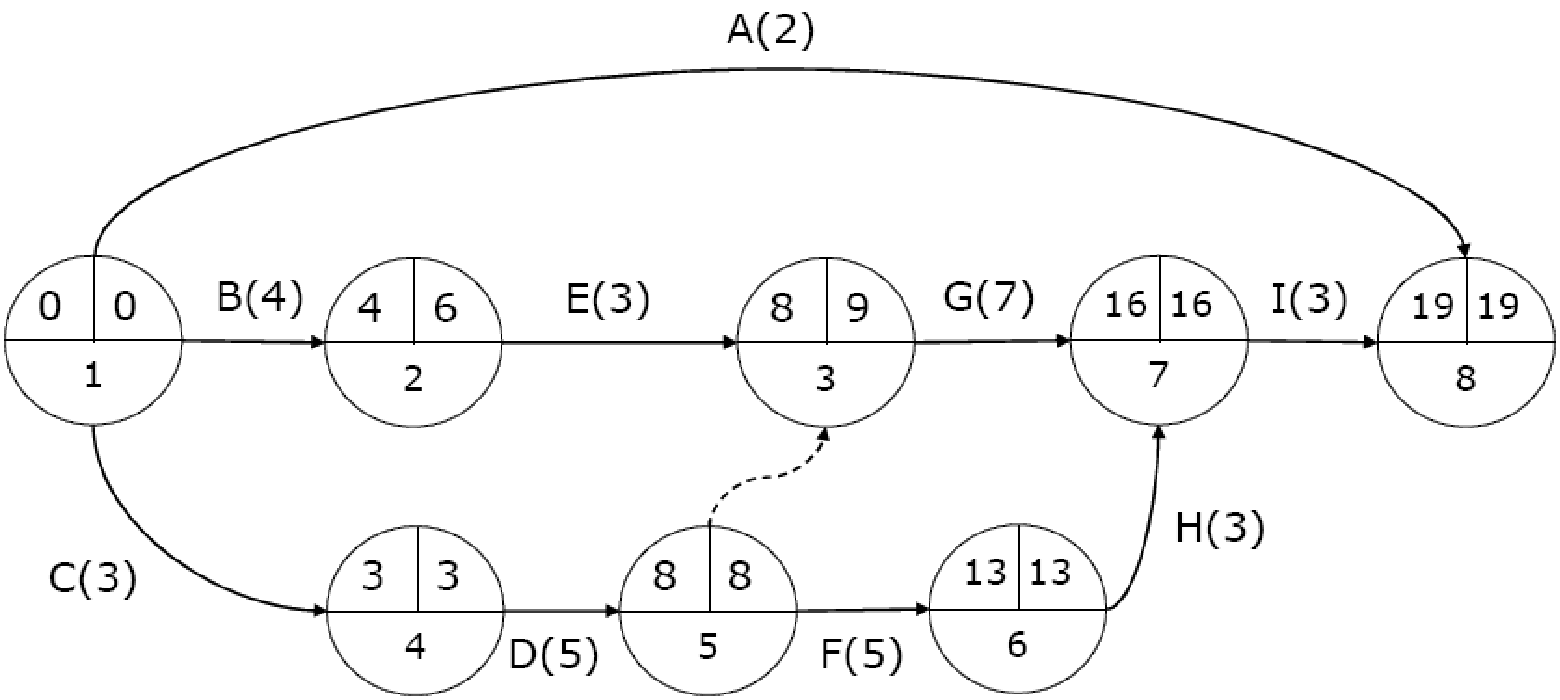
STAFFING & RE-SCHEDULING

- The obvious way to reduce the overall project duration, it is by reducing the duration of the critical activities.
- Crashing Critical Activities refers to reducing the duration of a critical activity by allocating more resources to it.
- The risk is that crashing activities may actually reduce productivity and increase costs.

STAFFING & RE-SCHEDULING

- **Example 2**

  Apply the PERT/CPM method and construct a GANTT chart for the following list of activities with precedence and duration.

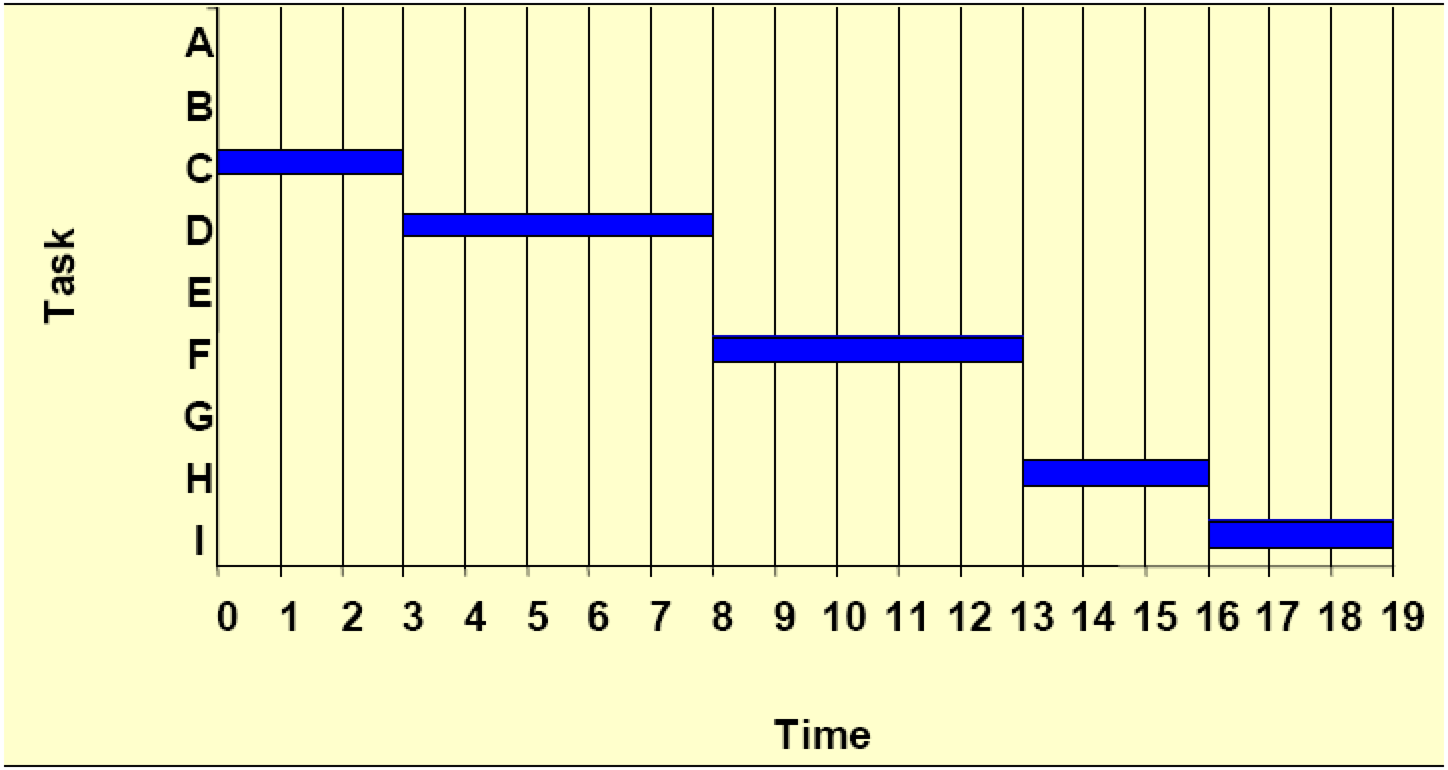


STAFFING & RE-SCHEDULING

Task	Precedence	Duration	ES	EF	LS	LF	Slack Time	Critical Task
A		2	0	2	17	19	17	No
B		4	0	4	2	6	2	No
C		3	0	3	0	3	0	Yes
D	C	5	3	8	3	8	0	Yes
E	B	3	4	7	6	9	2	No
F	D	5	8	13	8	13	0	Yes
G	D,E	7	8	15	9	16	1	No
H	F	3	13	16	13	16	0	Yes
I	G,H	3	16	19	16	19	0	Yes

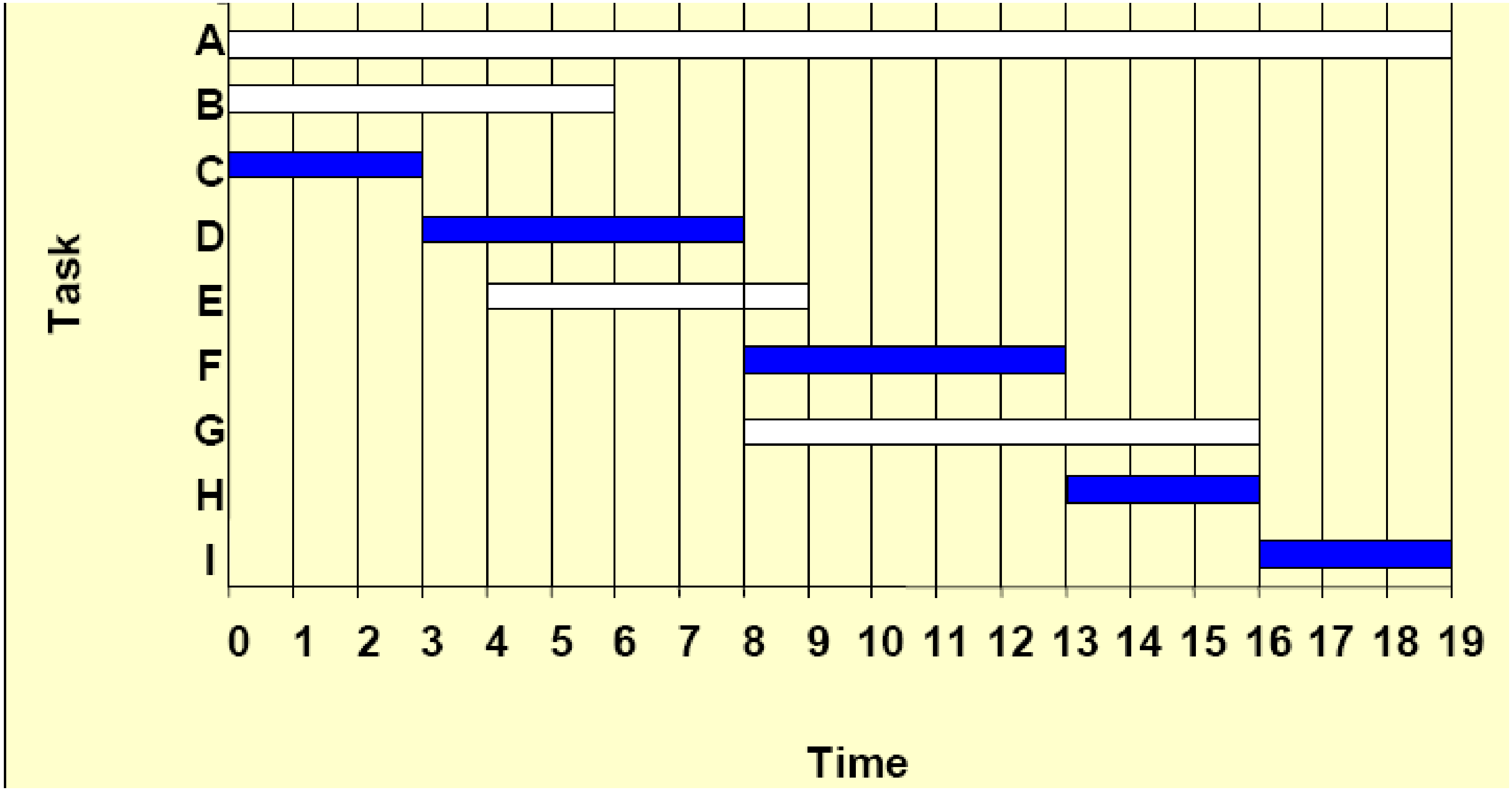
STAFFING & RE-SCHEDULING

Step 1. Schedule critical tasks.



STAFFING & RE-SCHEDULING

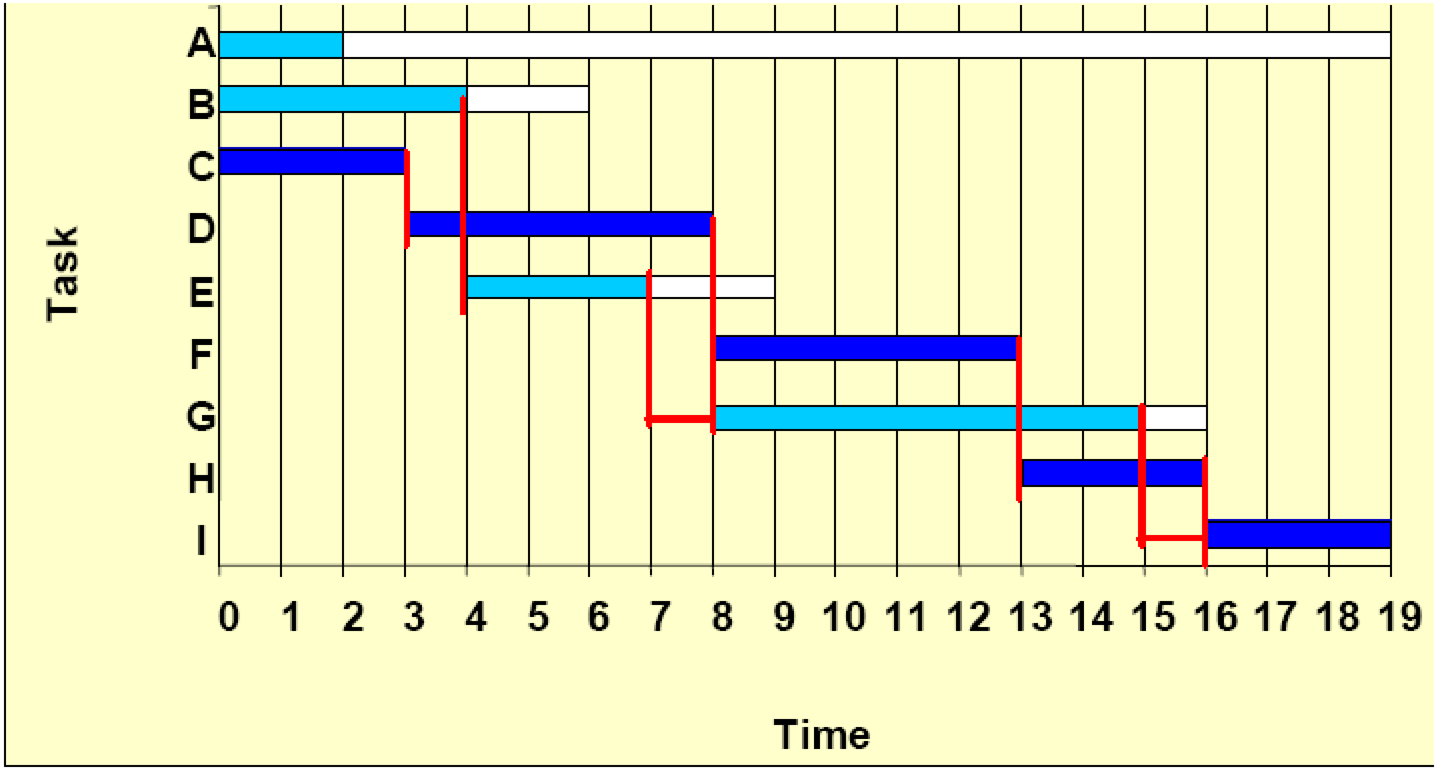
Step 2. Place time windows for non-critical tasks:



Staffing & Re-scheduling

Step 3. Schedule non-critical tasks:

Step 4. Indicate precedence relationships:



Project Management Software

- There are a number of project management software tools available to help in the planning and control of large software development projects.
    - E.g. MS Project is a CASE software tool for Project Management

- Most tools include functions to plan, schedule and control, *but* decision-making still has to be done by the project manager.

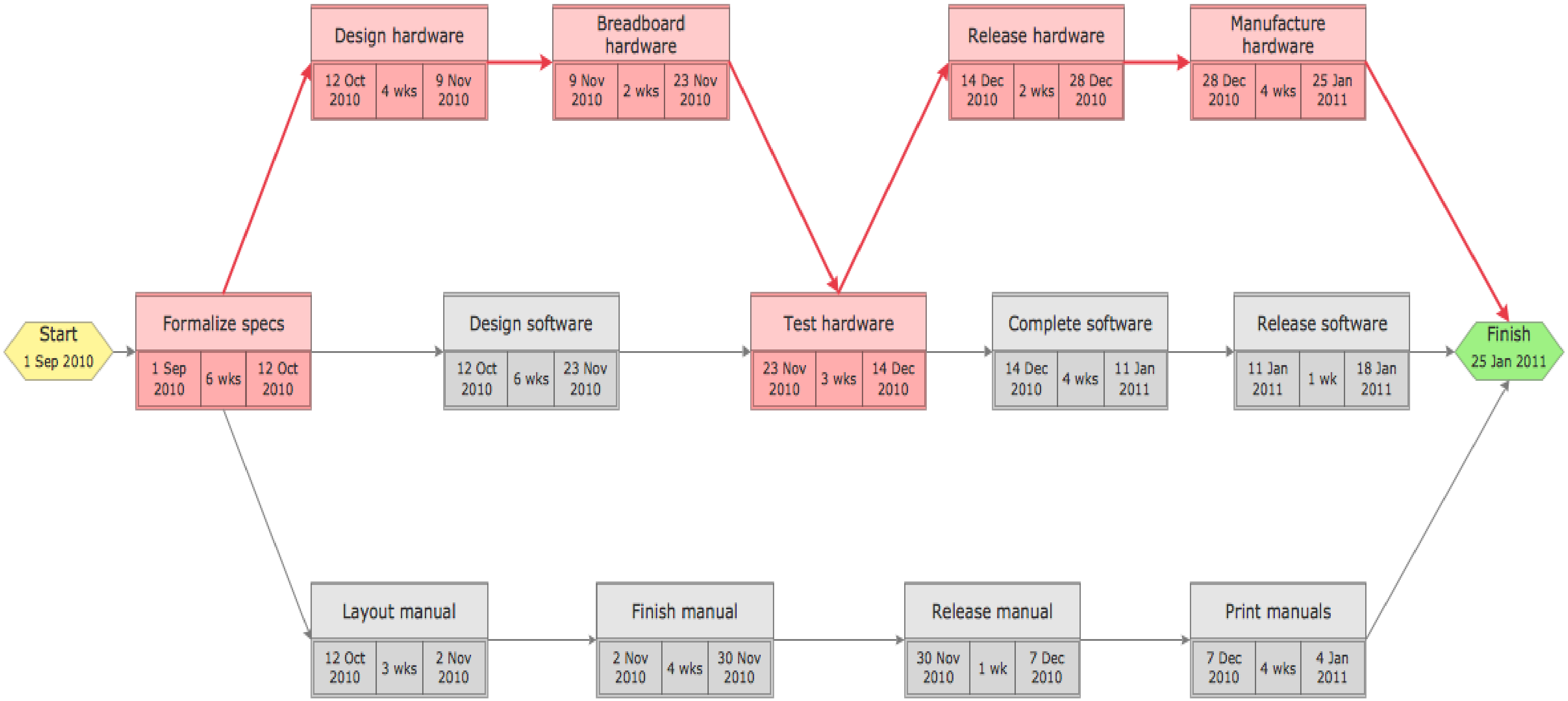
Project Management Software

- Benefits of project management software:
  - Calculate project schedule
  - Resource smoothing
  - Automatic generation of reports and charts

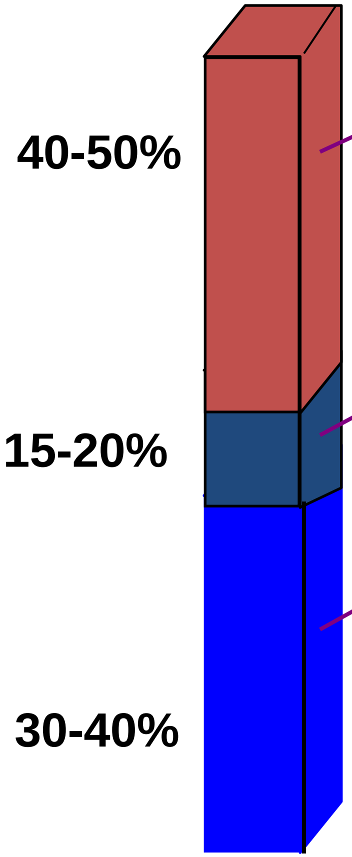
- Limitations of project management software
  - Allocation of resources to tasks
  - Estimation of tasks durations
  - Make decisions

Reading: (Kendall&Kendall, chapter 3), (Dennis &Wixom, chapter 3).

Other


Start
1 Sep 2010
Formalize specs
1 Sep 2010
6 wks
12 Oct 2010
Design hardware
12 Oct 2010
4 wks
9 Nov 2010
Breadboard hardware
9 Nov 2010
2 wks
23 Nov 2010
Design software
12 Oct 2010
6 wks
23 Nov 2010
Test hardware
23 Nov 2010
3 wks
14 Dec 2010
Release hardware
14 Dec 2010
2 wks
28 Dec 2010
Manufacture hardware
28 Dec 2010
4 wks
25 Jan 2011
Complete software
14 Dec 2010
4 wks
11 Jan 2011
Release software
11 Jan 2011
1 wk
18 Jan 2011
Layout manual
12 Oct 2010
3 wks
2 Nov 2010
Finish manual
2 Nov 2010
4 wks
30 Nov 2010
Release manual
30 Nov 2010
1 wk
7 Dec 2010
Print manuals
7 Dec 2010
4 wks
4 Jan 2011
Finish
25 Jan 2011

Effort Allocation



- “front end” activities
  - customer communication
  - analysis
  - design
  - review and modification
- construction activities
  - coding or code generation
- testing and installation
  - unit, integration
  - white-box, black box
  - regression

Task Set Refinement

1.1 **Concept scoping** determines the overall scope of the project.

is refined to

Task definition: Task 1.1 Concept Scoping

1.1.1 Identify need, benefits and potential customers;

1.1.2 Define desired output/control and input events that drive the application;

Begin Task 1.1.2

1.1.2.1 FTR: Review written description of need

FTR indicates that a formal technical review (Chapter 26) is to be conducted.

1.1.2.2 Derive a list of customer visible outputs/inputs

1.1.2.3 FTR: Review outputs/inputs with customer and revise as required;

endtask Task 1.1.2

1.1.3 Define the functionality/behavior for each major function;

Begin Task 1.1.3

1.1.3.1 FTR: Review output and input data objects derived in task 1.1.2;

1.1.3.2 Derive a model of functions/behaviors;

1.1.3.3 FTR: Review functions/behaviors with customer and revise as required;

endtask Task 1.1.3

1.1.4 Isolate those elements of the technology to be implemented in software;

1.1.5 Research availability of existing software;

1.1.6 Define technical feasibility;

1.1.7 Make quick estimate of size;

1.1.8 Create a Scope Definition;

endTask definition: Task 1.1

Progress on an OO Project-I

- *Technical milestone: OO analysis completed*
  - All classes and the class hierarchy have been defined and reviewed.
  - Class attributes and operations associated with a class have been defined and reviewed.
  - Class relationships (Chapter 8) have been established and reviewed.
  - A behavioral model (Chapter 8) has been created and reviewed.
  - Reusable classes have been noted.
- *Technical milestone: OO design completed*
  - The set of subsystems (Chapter 9) has been defined and reviewed.
  - Classes are allocated to subsystems and reviewed.
  - Task allocation has been established and reviewed.
  - Responsibilities and collaborations (Chapter 9) have been identified.
  - Attributes and operations have been designed and reviewed.
  - The communication model has been created and reviewed.

Progress on an OO Project-II

- *Technical milestone: OO programming completed*
  - Each new class has been implemented in code from the design model.
  - Extracted classes (from a reuse library) have been implemented.
  - Prototype or increment has been built.
- *Technical milestone: OO testing*
  - The correctness and completeness of OO analysis and design models has been reviewed.
  - A class-responsibility-collaboration network (Chapter 6) has been developed and reviewed.
  - Test cases are designed and class-level tests (Chapter 19) have been conducted for each class.
  - Test cases are designed and cluster testing (Chapter 19) is completed and the classes are integrated.
  - System level tests have been completed.

COMPUTING EARNED VALUE-I

- The *budgeted cost of work scheduled* (BCWS) is determined for each work task represented in the schedule.
  - $BCWS_i$ is the effort planned for work task *i*.
  - To determine progress at a given point along the project schedule, the value of BCWS is the sum of the $BCWS_i$ values for all work tasks that should have been completed by that point in time on the project schedule.
- The BCWS values for all work tasks are summed to derive the *budget at completion,* BAC. Hence,

$$BAC = \sum (BCWS_k) \text{ for all tasks } k$$

Computing Earned Value-II

- Next, the value for *budgeted cost of work performed* (BCWP) is computed.
  - The value for BCWP is the sum of the BCWS values for all work tasks that have actually been completed by a point in time on the project schedule.
- "the distinction between the BCWS and the BCWP is that the former represents the budget of the activities that were planned to be completed and the latter represents the budget of the activities that actually were completed." [Wil99]
- Given values for BCWS, BAC, and BCWP, important progress indicators can be computed:
  - Schedule performance index, SPI = BCWP/BCWS
  - Schedule variance, SV = BCWP – BCWS
  - SPI is an indication of the efficiency with which the project is utilizing scheduled resources.

COMPUTING EARNED VALUE-III

- Percent scheduled for completion = BCWS/BAC
  - provides an indication of the percentage of work that should have been completed by time *t*.
- Percent complete = BCWP/BAC
  - provides a quantitative indication of the percent of completeness of the project at a given point in time, *t*.
- *Actual cost of work performed,* ACWP, is the sum of the effort actually expended on work tasks that have been completed by a point in time on the project schedule. It is then possible to compute
  - Cost performance index, CPI = BCWP/ACWP
  - Cost variance, CV = BCWP – ACWP