بسم الله الرحمن الرحيم

# ENGINEERING MANAGEMENT

# (GE 404)

1

# LECTURE #5
# Scheduling Networks- AON

# Contents

(2)

- Objectives of the present lecture
- Activities versus events
- Duration versus Event
- Constraints
- Network Modeling Technique
- Steps in building a network model
- Early Start, Early Finish and
  Early Finish and Early Start Times
- Float Times
- Developing AON Network
- Problems
- Further reading

# **Objectives of the Present lecture**

- *To provide an overview of CPM based Network Modeling Techniques*

- *To discuss steps involved in drawing AON network diagrams*

- *To discuss how to calculate Early/Late Start/Finish Times*

- *To learn how to identify critical path in AON network diagrams*

# Activities versus Events

| Activities | Events |
|---|---|
| • Activities consume time. | • Events simply occur at a point in time |
| | • The point in time at which an activity can start may be considered an event |
| | • The point in time or instant at which an activity is completed can also be considered an event |

# Duration versus Event

| Duration | Event |
|---|---|
| • The duration of an activity is the period of time that will be consumed in completing a task<br><br>• The duration of the activity includes the time at which the activity begins and continues to the point in time at which it is completed. | • An event is the point in time or an instant at which the status of completion of a project or activity can be defined.<br><br>• The starting time for an activity, which is also an event, defines that point in time at which an activity can begin. |

# Constraints

- A number of different constraints must be acknowledged by the project team. The common constraints are:

  - Physical Constraints
  - Resource Constraints
  - Productivity Constraints
  - Safety Constraints
  - Financial Constraints

  - Environmental Constraints
  - Management Constraints
  - Contractual Constraints
  - Regulatory Constraints

# Impact of Constraints

- Introduction of excessive constraints in network logic can have the following impacts on a project:
  - Reduce scheduling flexibility
  - Lengthen project duration
  - Generally increase project cost
  - Confuse basic scheduling logic

# Network Modeling Technique—CPM

- CPM is the most popular network modeling technique
- In CPM technique, each activity is assigned a specific duration and calculations through the network provide a single, specific duration for the project as a whole
- CPM identifies those chains of activities (the critical paths) in the project that control how long the project will take
- There are two popular variations of CPM
  - Activity-on-Arrow (A-on-A), or an arrow diagram
    - This name is given to this traditional technique because activities are represented in the network as arrows or lines
  - Activity-on-Node or Precedence technique
    - This method defines the activities as boxes (nodes in the network), which are connected together by lines (links).
    - This technique has got popularity because of the availability of microcomputer software, which makes it very easy to use.

# Steps in Building a Network Model

- Define activities

- Order activities

- Establish activity relationships; draw a network diagram

- Determine quantities; assign durations to activities

- Assign resources and costs

- Calculate early and late start/finish times

- Compute float values and identify the critical path

- Schedule activity start/finish times

# Defining Activities

- Production/Construction Activities
  - These are activities that relate directly to the physical effort of creating the project
  - These activities use traditional resources of labor and materials

- Procurement Activities
  - These activities include arranging for the acquisition of materials, money, equipment, and workforce

- Management Activities
  - Activities such as preparing inspection reports, processing shop drawing approvals, tracking submittal approvals, developing as-built drawings, providing certifications on factory tests performed, and a variety of similar tasks.

# Ordering Activities

- For each identified activity, the following must be determined:
  - *Which activities must precede it?*
  - *Which activities must follow it?*
  - *Which activities can be concurrent with it?*

# Assigning Durations to Activities

- The duration of an activity is the estimated time that will be required to complete it

- Duration of the activity can be computed as:

$$\frac{\text{Qunatity of work}}{\left(\text{qty/crew hour}\right) \times \text{Number of crews}} = \text{crew} - \text{hours} = \text{Duration (days)}$$

- An alternative is to use historical Quantitative data from actual projects the firm has previously undertaken

# Early Start, Early Finish and Late Finish and Late Start Times

❑ The *"Early Start" (ES)* or "Earliest Start" of an activity is the earliest time that the activity can possibly start allowing for the time required to complete the preceding activities.

❑ The *"Early Finish" (EF)* or "Earliest Finish" of an activity is the earliest possible time that it can be completed and is determined by adding that activity's duration to its early start time.

❑ The *"Late Finish" (LF)* or "Latest Finish" of an activity is the very latest that it can finish and allow the entire project to be completed by a designated time or date.

❑ The *"Late Start" (LS)* or "Latest Start" of an activity is the latest possible time that it can be started if the project target completion date is to be met and is obtained by subtracting the activity's duration from its latest finish time.

# Calculation of Early and Late times

| Early | Late |
|---|---|

**Early**

- **Direction**: Proceed from project start to project finish, from **left to right**.

- **Name**: This process is called the "**forward pass**".

- **Assumption**: every activity will start as early as possible. That is to say, each activity will start just as soon as the last of its predecessors is finished.

- The **ES** value of each activity is determined first.

- The **EF** time is obtained by adding *the activity duration* to the ES time. *EF = ES + D*

- *In case of merge activities the earliest possible start time is equal to the latest (or largest) of the EF values of the immediately preceding activities.*

**Late**

- **Direction**: Proceed from project end to project start, from **right to left**.

- **Name**: This process is called the "**backward pass**".

- **Assumption**: Each activity finishes as late as possible without delaying project completion.

- The **LF** value of each activity is obtained first and is entered into lower right portion of the activity box.

- The **LS** is obtained by subtracting the activity duration from the LF value. *LS = LF - D*

- *In case of burst activities LF value is equal to the earliest (or smallest) of the LS times of the activities following.*

# Float Times —Free and Total Floats

- *Float* (or leeway) is a measure of the time available for a given activity above and beyond its estimated duration.

- The free float of an activity is found by subtracting its earliest finish time from the earliest start time of the activities directly following:

- $FF_i = Min. (ES_j) - EF_i$

- The **Total float** of an activity is obtained by subtracting its ES time from its LS time. Subtracting the EF from the LF gives the same result.

- **Total float (TF) = LS - ES = LF - EF**

- An activity with **zero** total float has **no spare time** and is, therefore, one of the operations that controls project completion time.

- Activities with zero total float are called **critical activities**.

# Free and Total Floats

| Free Float or activity Float | Total Float or Path  Float |
|---|---|
| *Free float* is the amount of time that an activity's completion time may be delayed **without affecting the earliest start of succeeding activity**<br><br>Free float is "owned" by an individual activity | *Total float* is the amount of time that an activity's completion may be delayed **without affecting the earliest start of any activity on the network critical path**<br><br>Total float is shared by all activities along a slack path<br><br>Total path float time for activity (i-j) is the total float associated with a path |

# Activity Box
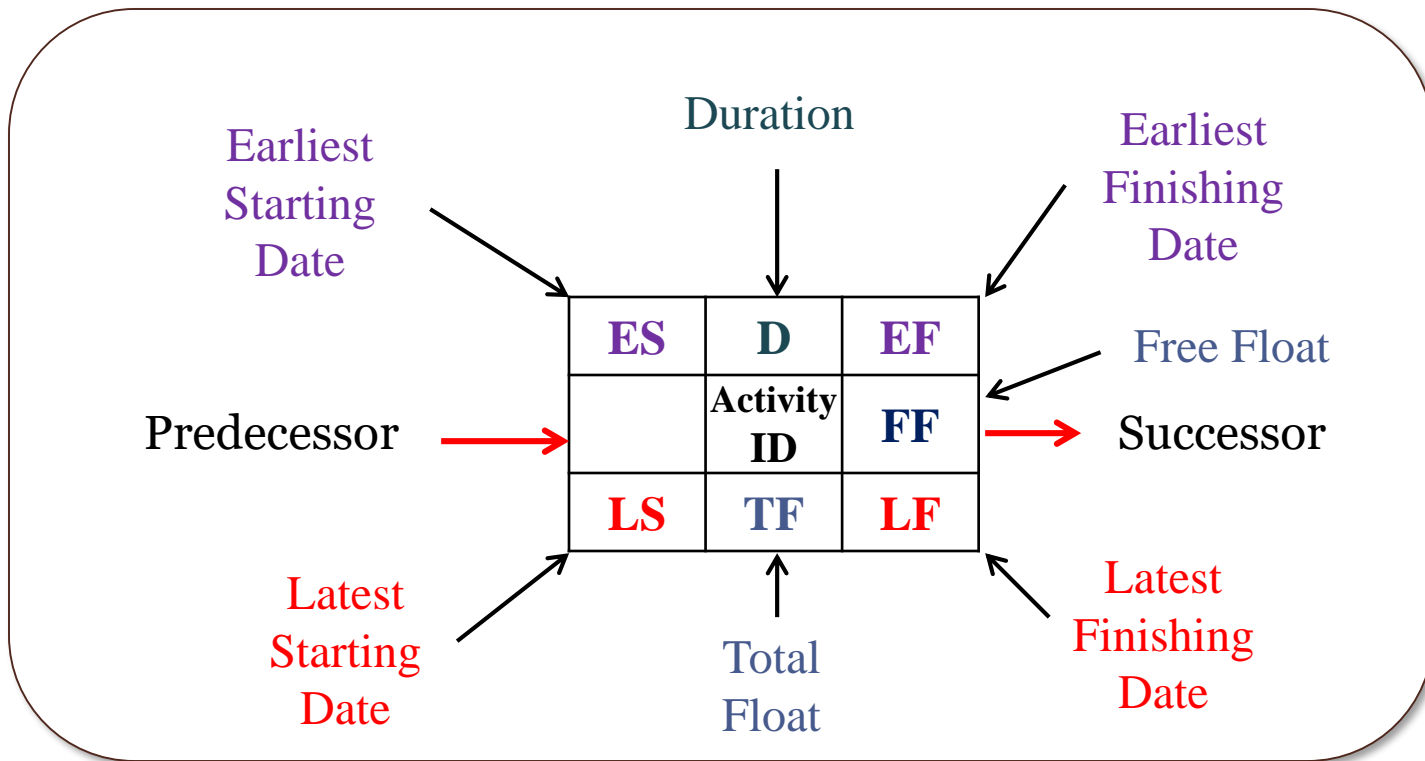
The format for activity box is not standardized, so one need only to adopt a format that is comfortable for a particular application

# Critical Activity and Critical Path

- Critical activity is quickly identified as one whose **two start times** at the left of the activity box **are equal**. Also, **equal** are the **two finish times** at the right of the activity box.

- The critical activities **must form a continuous path from project beginning to project end**, this chain of critical activities is called the Critical Path.

- The critical path is the **longest path** in the network.

- The critical path is normally indicated on the diagram in some distinctive way such as with colors, heavy lines, or double lines.

- **Any delay in the finish date of a critical activity**, for whatever reason, automatically **prolongs project completion** by the same amount.

# **Subcritical paths**

○ Subcritical paths have varying degree of path float and hence depart from criticality by varying amounts

○ Subcritical paths can be found in the following way:
  ✗ Sort the activities in the network by their path float, placing those activities with a common path float in the same group
  ✗ Order the activities within a group by early start time
  ✗ Order the groups according to the magnitude of their path float, small values first

# Activity-on-Node (AON) Network

- The most common type of network schedule in use today is the Activity-on-Node
- Activities are represented by nodes, drawn in any desired shape
- Lines represent "Activity links," used to represent dependencies between activities
- The principal advantage of the activity on node network is that it eliminates the need for dummies
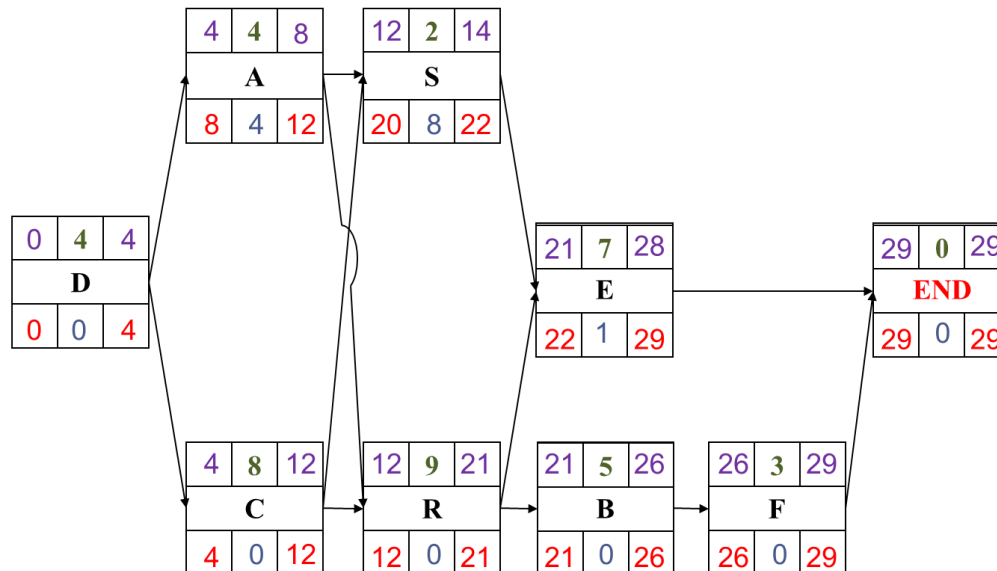


The diagram is "read" from left to right

# Developing AON Network

- The general developing of a network is from start to finish, from project beginning on the *left* to project completion on the *right*.

- A horizontal diagram format is the standard format

- The sequential relationship of one activity to another is shown by the dependency lines between them.

- The length of the lines between activities has no significance.

- Arrowheads are not always shown on the dependency lines because of the obvious left to right flow of time.

- **Dependency lines that go backward from one activity to another (looping) should not be used.**

- *Crossovers* **occur when one dependency line must cross over another to satisfy job logic.**
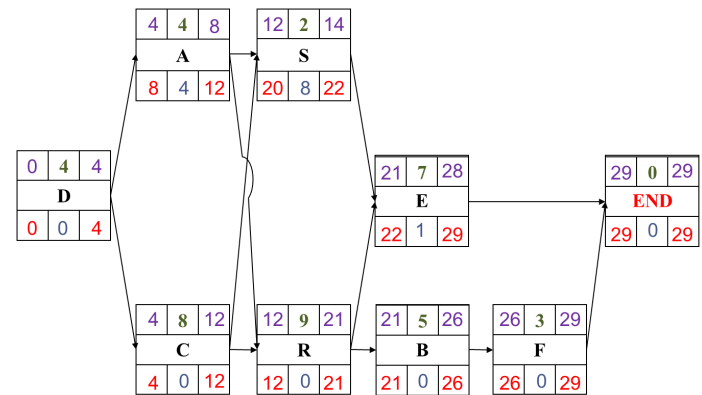
# Note

- Each activity in the network must be preceded either by the start of the project or by the completion of a previous activity.

- Each path through the network must be **continuous** with no gaps, discontinuities, or dangling activities.

- All activities must have at least one activity following, except the activity that terminates the project.

- Each activity should have a unique numerical designation (activity code).

- Activity code is shown in the central part of the activity box, with the numbering proceeding generally from project start to finish.

# Problem-1

The activity list shown below represents the activities, the job logic and the activities' durations of a small project. Draw an activity on node network to represent the project.

| Activity | Depends on | Duration (days) |
|----------|------------|-----------------|
| A | D | 4 |
| B | R | 5 |
| C | D | 8 |
| E | R, S | 7 |
| F | B, C | 3 |
| D | None | 4 |
| S | A, C | 2 |
| R | A, C | 9 |

| ES | t | EF |
|----|---|----|
| | Activity | |
| LS | TF | LF |

# Solution
## Step-1: Draw Activity diagram

| Activity | Depends on | Duration (days) |
|---|---|---|
| A | D | 4 |
| B | R | 5 |
| C | D | 8 |
| E | R, S | 7 |
| F | B, C | 3 |
| D | None | 4 |
| S | A, C | 2 |
| R | A, C | 9 |

| ES | t | EF |
|---|---|---|
| | Activity | |
| LS | TF | LF |

*When you have to draw an activity diagram. Pick the first activity and then see on which it depends and then see on which these (i.e. predecessors) depend. For example, to draw activity A, see activity D then see on which activity D depends and so on.*



| | 4 | |
|---|---|---|
| | **A** | |
| | | |

| | 2 | |
|---|---|---|
| | **S** | |
| | | |

| | 4 | |
|---|---|---|
| | **D** | |
| | | |

| | 7 | |
|---|---|---|
| | **E** | |
| | | |

| | 0 | |
|---|---|---|
| | **END** | |
| | | |

| | 8 | |
|---|---|---|
| | **C** | |
| | | |

| | 9 | |
|---|---|---|
| | **R** | |
| | | |

| | 5 | |
|---|---|---|
| | **B** | |
| | | |

| | 3 | |
|---|---|---|
| | **F** | |
| | | |
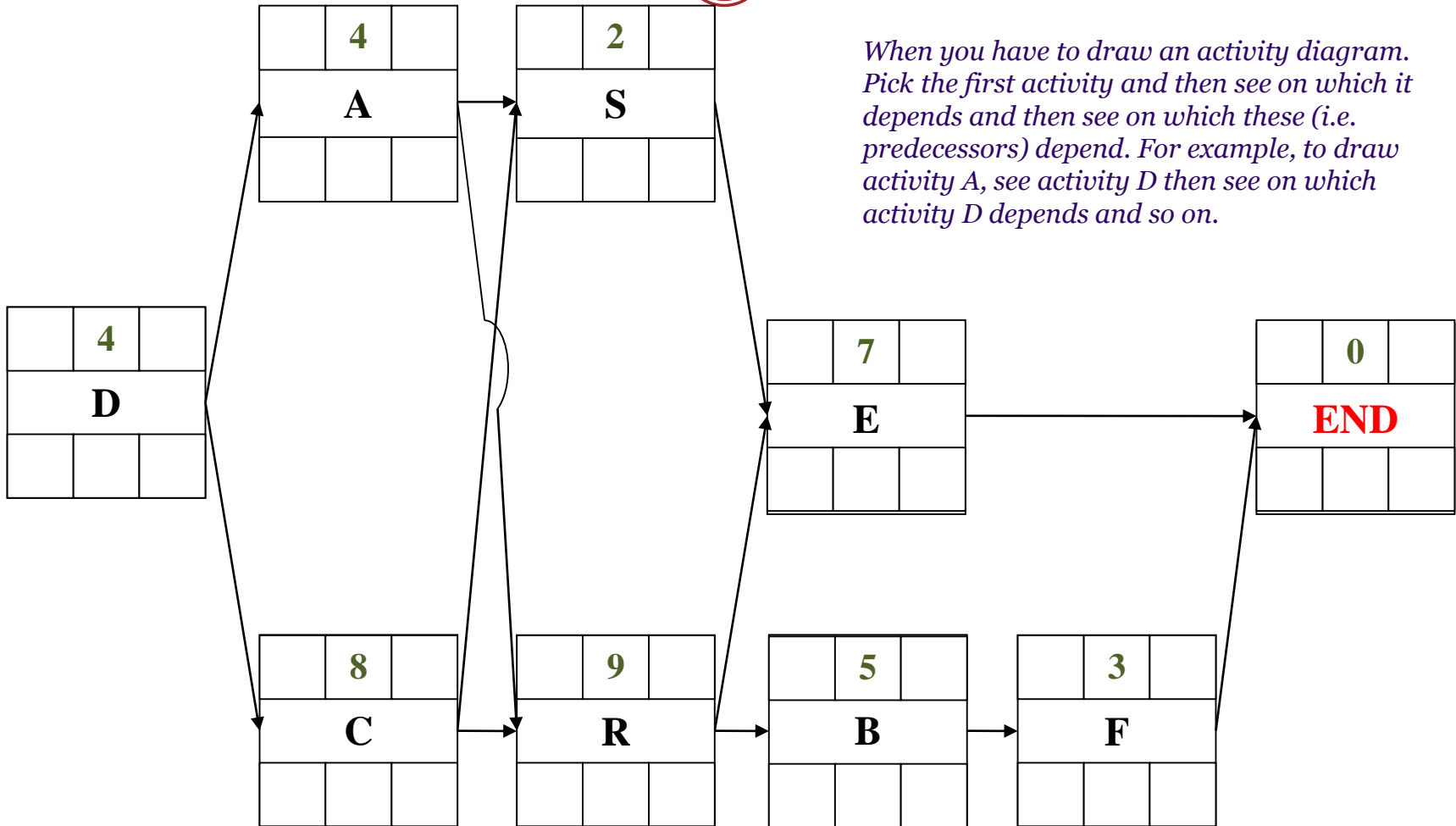
$$EF = ES + D$$
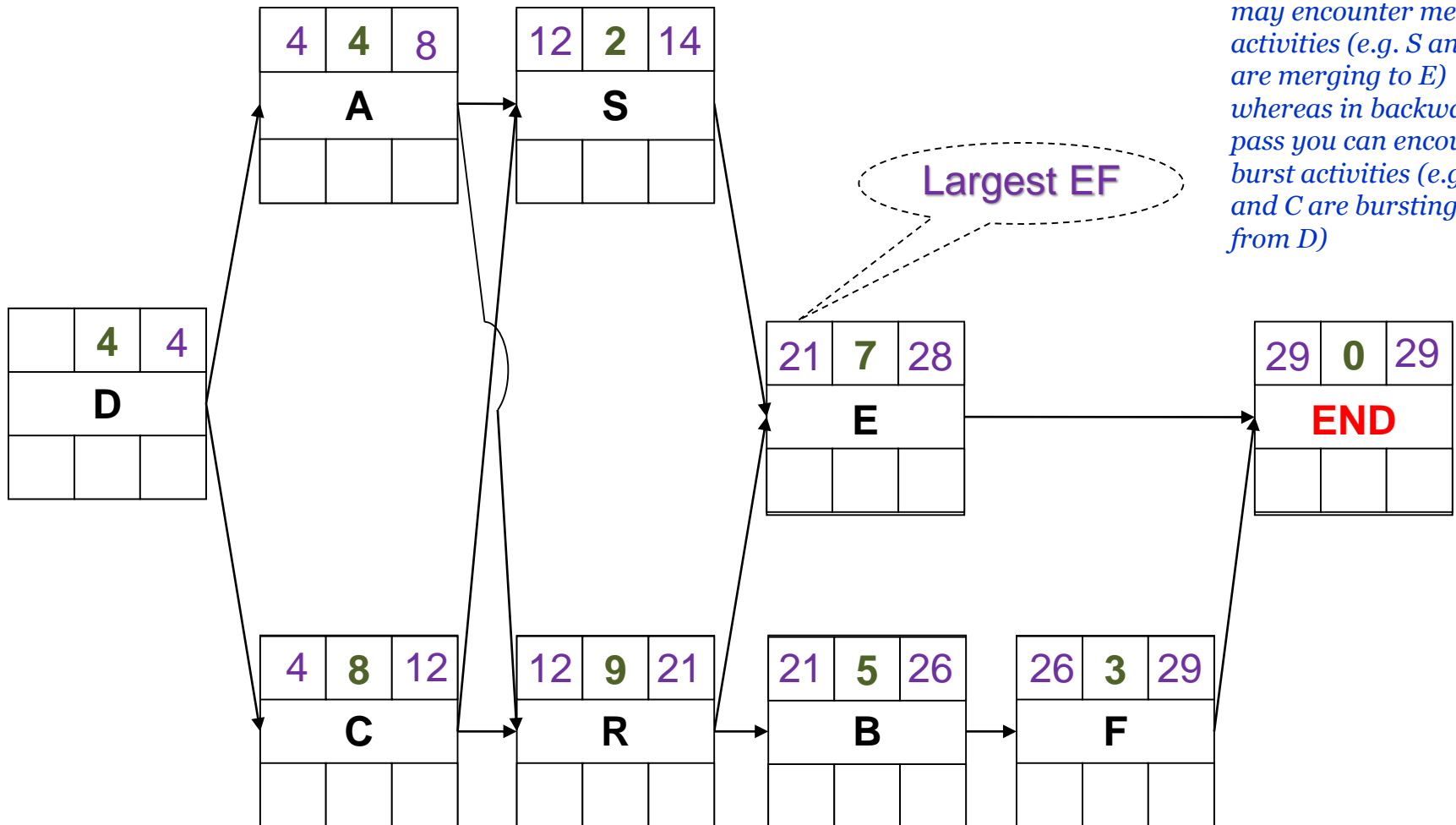
*In forward pass you may encounter merge activities (e.g. S and R are merging to E) whereas in backward pass you can encounter burst activities (e.g. A and C are bursting from D)*

| 4 | 4 | 8 |
|---|---|---|
| | A | |
| | | |

| 12 | 2 | 14 |
|----|---|----|
| | S | |
| | | |

Largest EF

| | 4 | 4 |
|---|---|---|
| | D | |
| | | |

| 21 | 7 | 28 |
|----|---|----|
| | E | |
| | | |

| 29 | 0 | 29 |
|----|---|----|
| | **END** | |
| | | |

| 4 | 8 | 12 |
|---|---|----|
| | C | |
| | | |

| 12 | 9 | 21 |
|----|---|----|
| | R | |
| | | |

| 21 | 5 | 26 |
|----|---|----|
| | B | |
| | | |

| 26 | 3 | 29 |
|----|---|----|
| | F | |
| | | |

# Step-3: Calculate LS and LF

$$LS = LF - D$$

| | | |
|---|---|---|
| 4 | 4 | 8 |
| | A | |
| 8 | | 12 |

| | | |
|---|---|---|
| 12 | 2 | 14 |
| | S | |
| 20 | | 22 |

| | | |
|---|---|---|
| 0 | 4 | 4 |
| | D | |
| 0 | | 4 |

| | | |
|---|---|---|
| 21 | 7 | 28 |
| | E | |
| 22 | | 29 |

| | | |
|---|---|---|
| 29 | 0 | 29 |
| | END | |
| 29 | | 29 |

| | | |
|---|---|---|
| 4 | 8 | 12 |
| | C | |
| 4 | | 12 |

| | | |
|---|---|---|
| 12 | 9 | 21 |
| | R | |
| 12 | | 21 |

| | | |
|---|---|---|
| 21 | 5 | 26 |
| | B | |
| 21 | | 26 |

| | | |
|---|---|---|
| 26 | 3 | 29 |
| | F | |
| 26 | | 29 |
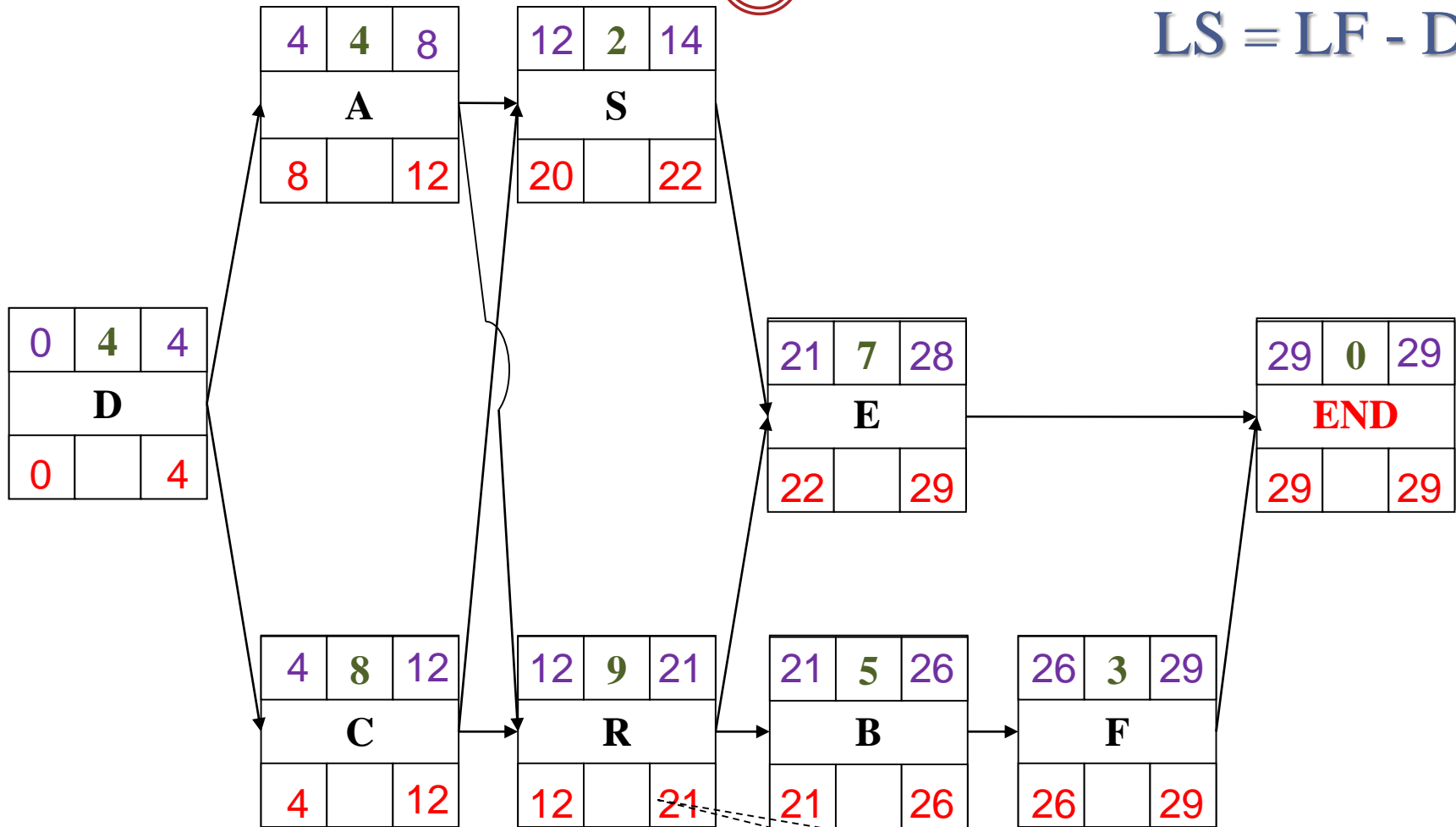
Smallest LS

# Step-4: Calculate Total Float

**Total float (TF) = LS - ES = LF - EF**



| | | |
|---|---|---|
| 4 | 4 | 8 |
| | A | |
| 8 | 4 | 12 |

| | | |
|---|---|---|
| 12 | 2 | 14 |
| | S | |
| 20 | 8 | 22 |

| | | |
|---|---|---|
| 0 | 4 | 4 |
| | D | |
| 0 | 0 | 4 |

| | | |
|---|---|---|
| 21 | 7 | 28 |
| | E | |
| 22 | 1 | 29 |

| | | |
|---|---|---|
| 29 | 0 | 29 |
| | END | |
| 29 | 0 | 29 |

| | | |
|---|---|---|
| 4 | 8 | 12 |
| | C | |
| 4 | 0 | 12 |

| | | |
|---|---|---|
| 12 | 9 | 21 |
| | R | |
| 12 | 0 | 21 |

| | | |
|---|---|---|
| 21 | 5 | 26 |
| | B | |
| 21 | 0 | 26 |

| | | |
|---|---|---|
| 26 | 3 | 29 |
| | F | |
| 26 | 0 | 29 |

| 4 | **4** | 8 |
|---|---|---|
| | A | |
| 8 | 4 | 12 |

| 12 | **2** | 14 |
|---|---|---|
| | S | |
| 20 | 8 | 22 |

| 0 | **4** | 4 |
|---|---|---|
| | D | |
| 0 | 0 | 4 |

| 21 | **7** | 28 |
|---|---|---|
| | E | |
| 22 | 1 | 29 |

| 29 | **0** | 29 |
|---|---|---|
| | **END** | |
| 29 | 0 | 29 |

| 4 | **8** | 12 |
|---|---|---|
| | C | |
| 4 | 0 | 12 |

| 12 | **9** | 21 |
|---|---|---|
| | R | |
| 12 | 0 | 21 |

| 21 | **5** | 26 |
|---|---|---|
| | B | |
| 21 | 0 | 26 |

| 26 | **3** | 29 |
|---|---|---|
| | F | |
| 26 | 0 | 29 |
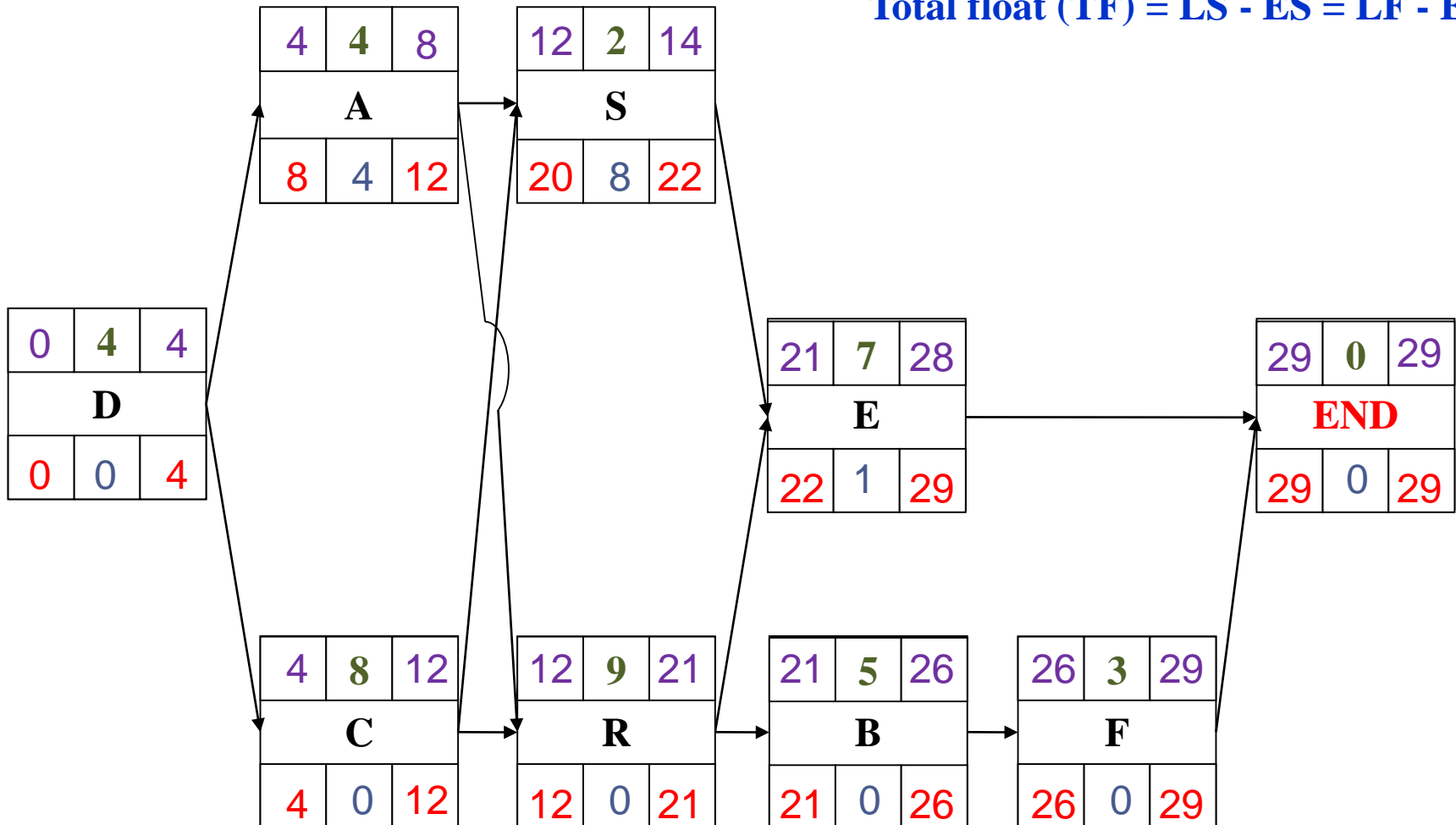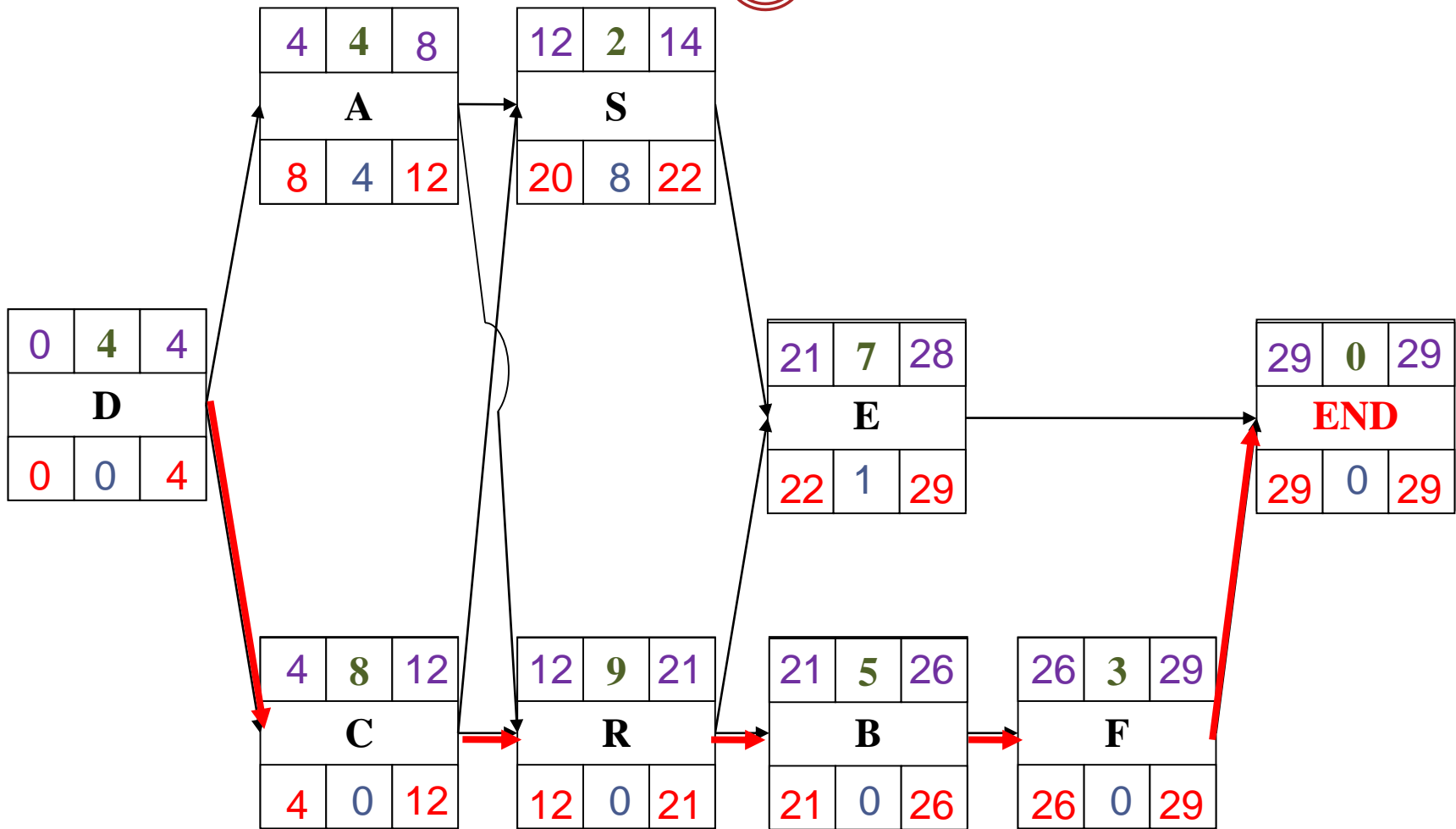
# Problem-2

Draw AON diagram to represent the following project. Calculate occurrence times of events (i.e. activity times), and activity and path floats. Also determine the critical path and the degree of criticality of other float paths.

| Activity | Preceding Activity | Time (days) |
|----------|--------------------|-------------|
| A | None | 5 |
| B | A | 7 |
| C | A | 4 |
| D | A | 8 |
| E | B | 6 |
| F | D | 8 |
| G | E, C, F | 3 |
| H | G | 4 |
| I | G | 6 |

| ES | t | EF |
|----|---|----|
| | Activity | |
| LS | TF | LF |
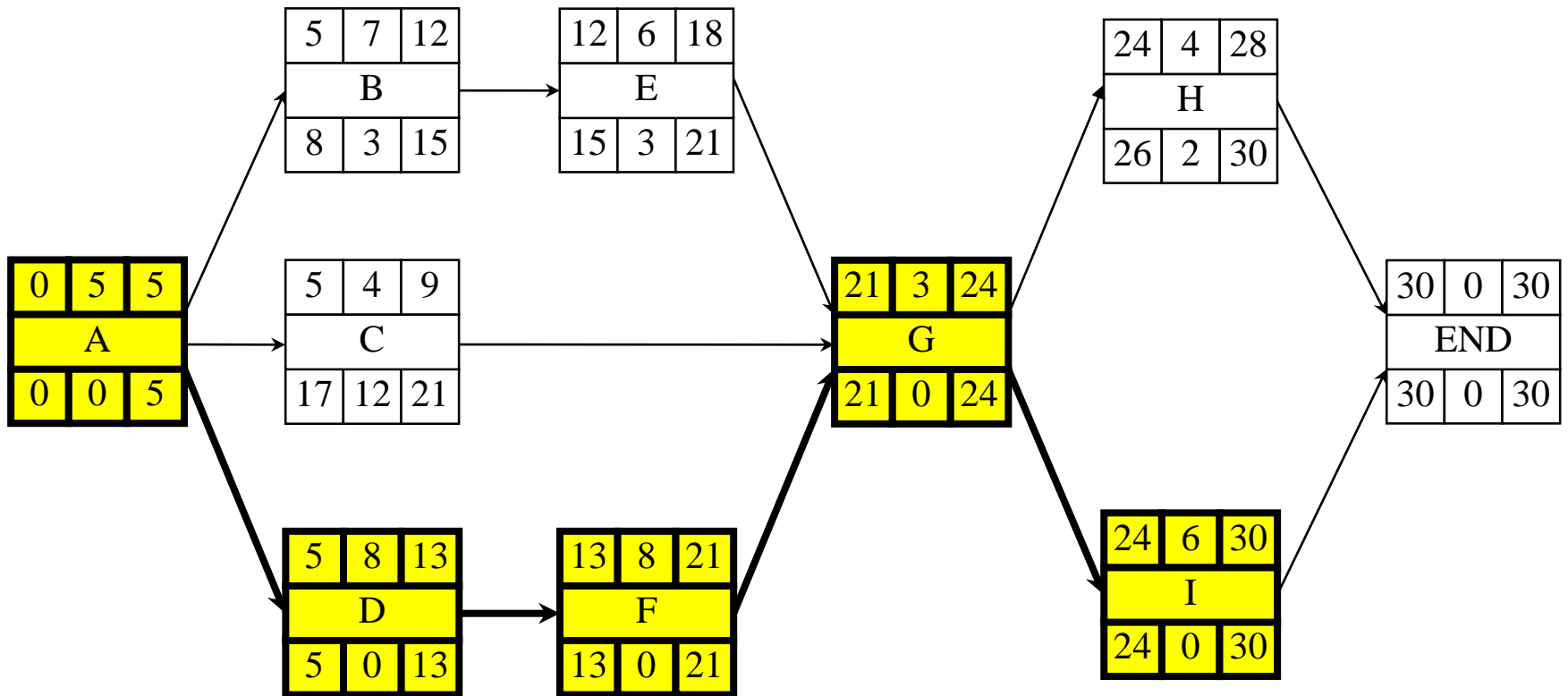
# Step 2: Activity times and activity floats

| Activity | ES | EF | LF | LS | TF | FF |
|----------|----|----|----|----|----|----|
| A | 0 | 5 | 5 | 0 | 0 | 0 |
| B | 5 | 12 | 15 | 8 | 3 | 0 |
| C | 5 | 9 | 21 | 17 | 12 | 12 |
| D | 5 | 13 | 13 | 5 | 0 | 0 |
| E | 12 | 18 | 21 | 15 | 3 | 3 |
| F | 13 | 21 | 21 | 13 | 0 | 0 |
| G | 21 | 24 | 24 | 21 | 0 | 0 |
| H | 24 | 28 | 30 | 26 | 2 | 2 |
| I | 24 | 30 | 30 | 24 | 0 | 0 |

# Step 3: Critical path and subcritical paths

| Activity | ES | EF | LF | LS | TF | Criticality |
|:---:|:---:|:---:|:---:|:---:|:---:|:---:|
| A | 0 | 5 | 5 | 0 | 0 | Critical Path |
| D | 5 | 13 | 13 | 5 | 0 | |
| F | 13 | 21 | 21 | 13 | 0 | |
| G | 21 | 24 | 24 | 21 | 0 | |
| I | 24 | 30 | 30 | 24 | 0 | |
| H | 24 | 28 | 30 | 26 | 2 | a "near critical" path |
| B | 5 | 12 | 15 | 8 | 3 | Third most critical path |
| E | 12 | 18 | 21 | 15 | 3 | |
| C | 5 | 9 | 21 | 17 | 12 | Path having most float |

# Problem-3

The activity list shown below represents the activities, the job logic and the activities' durations of a small project. Draw an activity on node network to represent the project.

| Activity | Description | Immediate Predecessors | Time (weeks) |
|---|---|---|---|
| A | Build internal components | — | 2 |
| B | Modify roof and floor | — | 3 |
| C | Construct collection stack | A | 2 |
| D | Pour concrete and install frame | A, B | 4 |
| E | Build high-temperature burner | C | 4 |
| F | Install pollution control system | C | 3 |
| G | Install air pollution device | D, E | 5 |
| H | Inspect and test | F, G | 2 |

| ES | t | EF |
|---|---|---|
| | Activity | |
| LS | TF | LF |

# Solution
## Step-1: Draw Activity diagram

| Activity | Description | Immediate Predecessors | Time (weeks) |
|---|---|---|---|
| A | Build internal components | — | 2 |
| B | Modify roof and floor | — | 3 |
| C | Construct collection stack | A | 2 |
| D | Pour concrete and install frame | A, B | 4 |
| E | Build high-temperature burner | C | 4 |
| F | Install pollution control system | C | 3 |
| G | Install air pollution device | D, E | 5 |
| H | Inspect and test | F, G | 2 |

# Step-2: Calculate ES and EF and Project duration

$$LS = LF - D$$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|

**A**: 0 | 2 | 2 ; 0 | | 2

**C**: 2 | 2 | 4 ; 2 | | 4

**F**: 4 | 3 | 7 ; 10 | | 13

**Start**: 0 | 0 | 0 ; 0 | | 0

**E**: 4 | 4 | 8 ; 4 | | 8

**H**: 13 | 2 | 15 ; 13 | | 15

**B**: 0 | 3 | 3 ; 1 | | 4

**D**: 3 | 4 | 7 ; 4 | | 8

**G**: 8 | 5 | 13 ; 8 | | 13

LF = Min(LS of activities 4,10)

LF = EF of Project
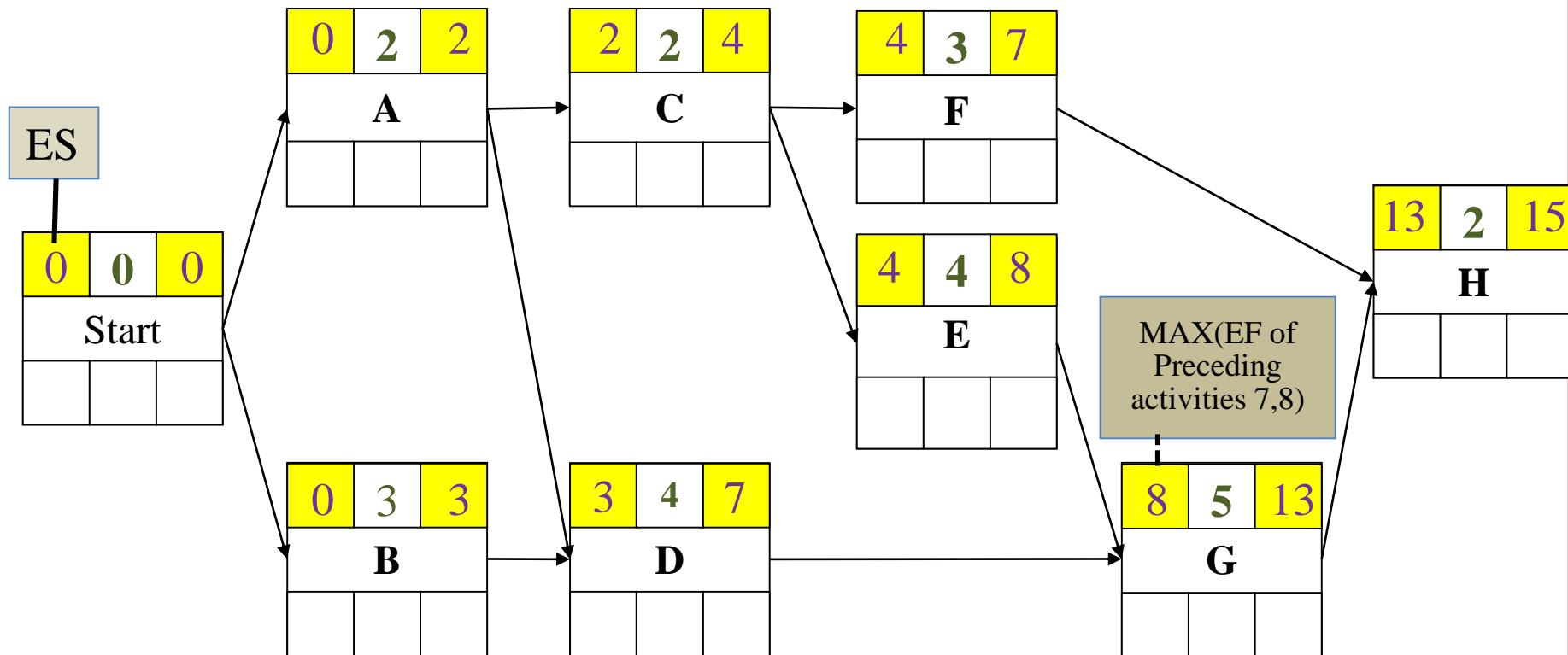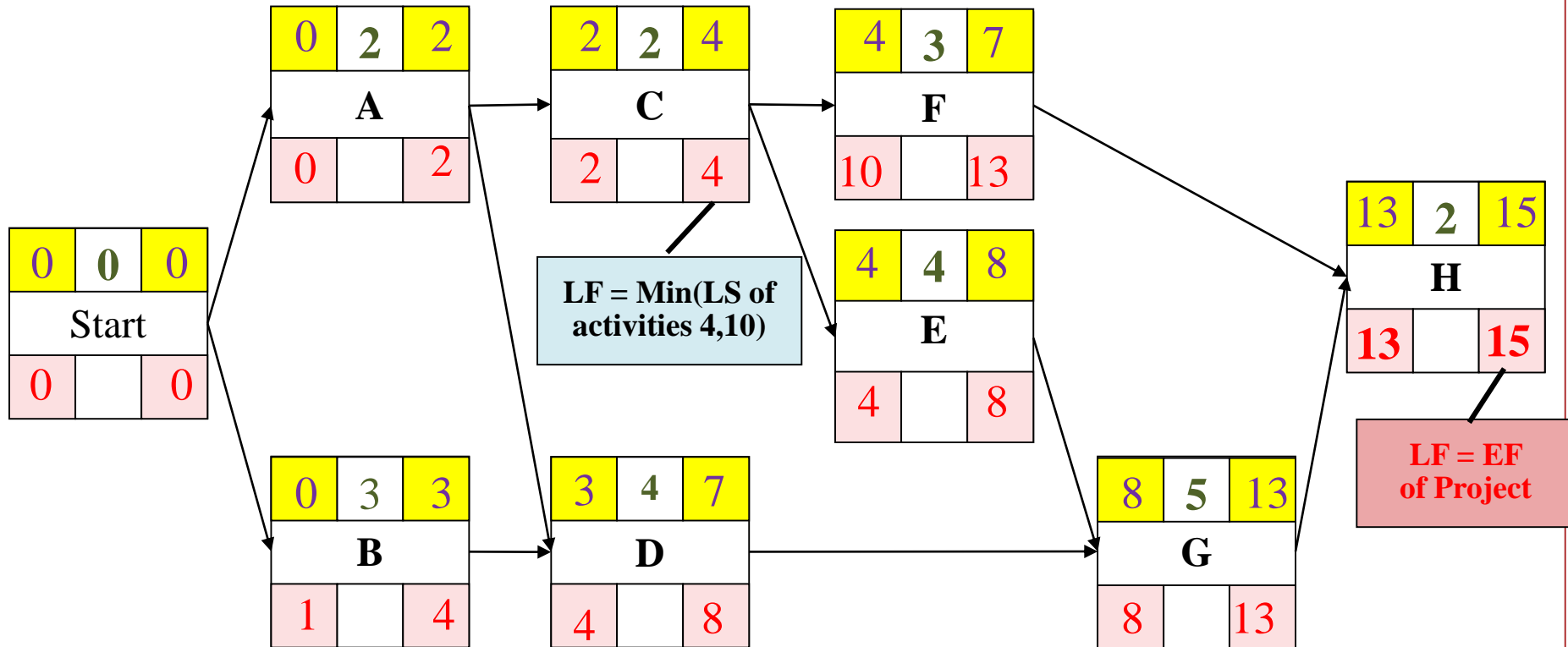
# Step-4: Calculate Total Float

**Total float (TF) = LS - ES = LF - EF**

*Slack = LS – ES    or*
*Slack = LF – EF*



| 0 | 2 | 2 |
|---|---|---|
| | A | |
| 0 | 0 | 2 |

| 2 | 2 | 4 |
|---|---|---|
| | C | |
| 2 | 0 | 4 |

| 4 | 3 | 7 |
|---|---|---|
| | F | |
| 10 | 6 | 13 |

| 0 | 0 | 0 |
|---|---|---|
| | Start | |
| 0 | 0 | 0 |

| 4 | 4 | 8 |
|---|---|---|
| | E | |
| 4 | 0 | 8 |

| 13 | 2 | 15 |
|---|---|---|
| | H | |
| 13 | 0 | 15 |

| 0 | 3 | 3 |
|---|---|---|
| | B | |
| 1 | 1 | 4 |

| 3 | 4 | 7 |
|---|---|---|
| | D | |
| 4 | 1 | 8 |

| 8 | 5 | 13 |
|---|---|---|
| | G | |
| 8 | 0 | 13 |

**LF = EF
of Project**
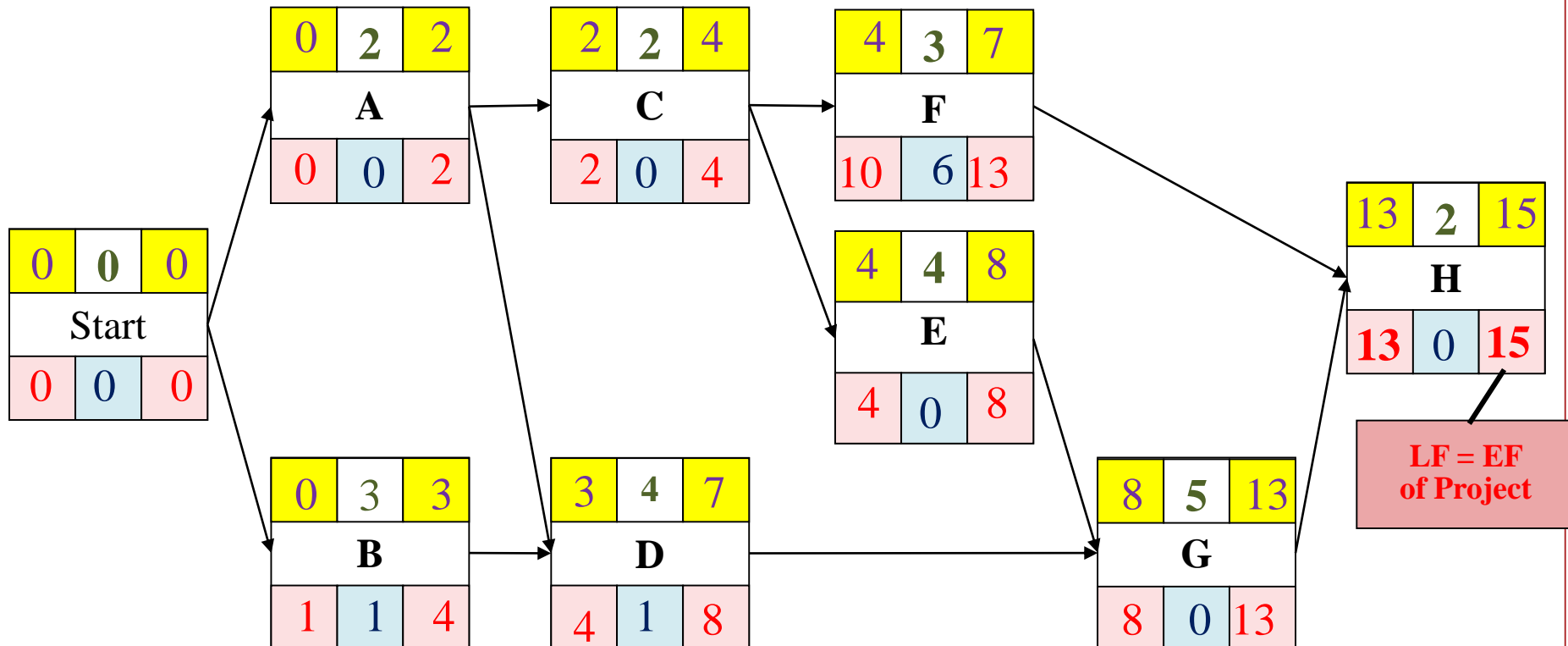
# Step-5: Identify Critical Path
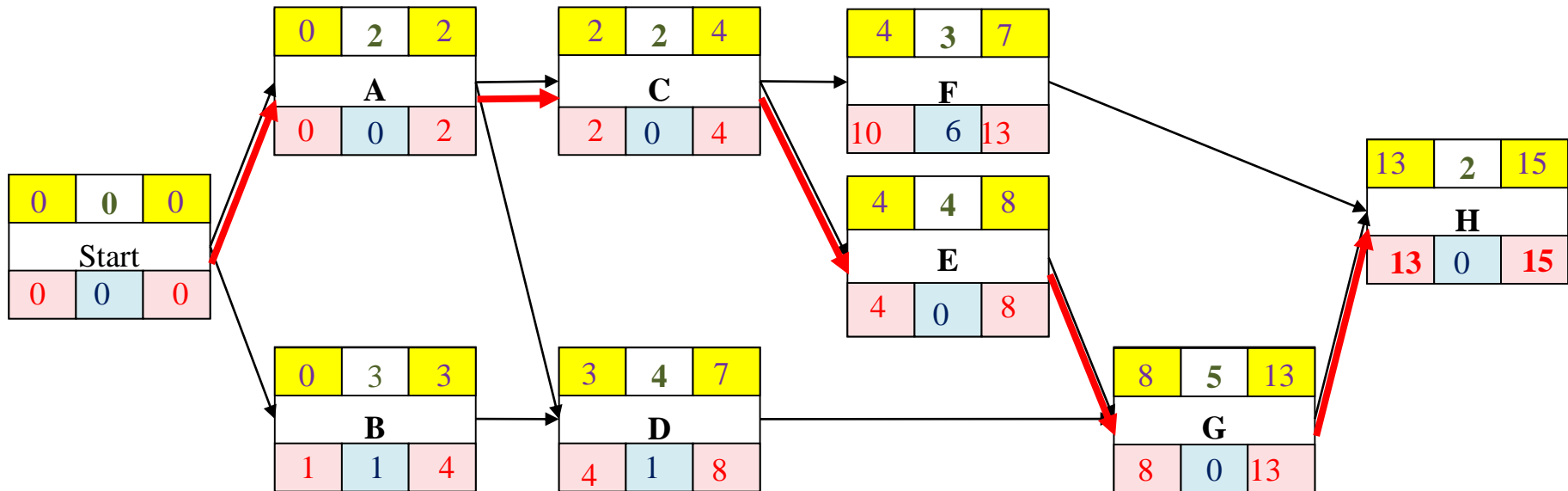
# Step-6: Calculate Slack Time (Float Time)

| Activity | Earliest Start ES | Earliest Finish EF | Latest Start LS | Latest Finish LF | Slack LS − ES | On Critical Path |
|----------|----|----|----|----|----|-----|
| A | 0 | 2 | 0 | 2 | 0 | Yes |
| B | 0 | 3 | 1 | 4 | 1 | No |
| C | 2 | 4 | 2 | 4 | 0 | Yes |
| D | 3 | 7 | 4 | 8 | 1 | No |
| E | 4 | 8 | 4 | 8 | 0 | Yes |
| F | 4 | 7 | 10 | 13 | 6 | No |
| G | 8 | 13 | 8 | 13 | 0 | Yes |
| H | 13 | 15 | 13 | 15 | 0 | Yes |

# GANTT CHART SCHEDULE
## (ES –EF )

| ACTIVITY | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | Build internal components | ■ | ■ | | | | | | | | | | | | | | |
| C | Construct collection stack | | | ■ | ■ | | | | | | | | | | | | |
| E | Build high-temperature burner | | | | | ■ | ■ | ■ | ■ | | | | | | | | |
| G | Install air pollution device | | | | | | | | | ■ | ■ | ■ | ■ | ■ | | | |
| H | Inspect and test | | | | | | | | | | | | | | ■ | ■ | |
| B | Modify roof and floor | ■ | ■ | ■ | | | | | | | | | | | | | |
| D | Pour concrete and install frame | | | | ■ | ■ | ■ | ■ | | | | | | | | | |
| F | Install pollution control system | | | | | ■ | ■ | ■ | | | | | | | | | |

## GANTT CHART SCHEDULE
### (LS –LF )

| ACTIVITY | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | Build internal components | ■ | ■ | | | | | | | | | | | | | | |
| C | Construct collection stack | | | ■ | ■ | | | | | | | | | | | | |
| E | Build high-temperature burner | | | | | ■ | ■ | ■ | ■ | | | | | | | | |
| G | Install air pollution device | | | | | | | | | ■ | ■ | ■ | ■ | ■ | | | |
| H | Inspect and test | | | | | | | | | | | | | | ■ | ■ | |
| B | Modify roof and floor | | ■ | ■ | ■ | | | | | | | | | | | | |
| D | Pour concrete and install frame | | | | | ■ | ■ | ■ | ■ | | | | | | | | |
| F | Install pollution control system | | | | | | | | | | | ■ | ■ | ■ | | | |

# Calendar Date Schedule

- Activity times (ES, EF, LS, LF) obtained from previous calculations are expressed in terms of **expired** working days.

- For purposes of project directing, monitoring and control, it is necessary to convert these times to calendar dates on which each activity is expected to start and finish.

- This is done with the aid of a calendar on which the working days are numbered consecutively, starting with number 1 on the anticipated start date and skipping weekends and holidays.

# Advantages and Disadvantages of Network Diagram

## *Advantages*

- Show precedence well
- Reveal interdependencies not shown in other techniques
- Ability to calculate critical path
- Ability to perform "what if" exercises

## *Disadvantages*

- Default model assumes resources are unlimited
- You need to incorporate this yourself (Resource Dependencies) when determining the "real" Critical Path
- Difficult to follow on large projects

# Note (a Constraint)

- **Question:**
- The last activity in a chain has no successor (there is only the project end date) and this activity has some positive number as a Total Float value.
- What should be the Free Float value for this activity ?
- **Answer:**
- Good scheduling practice requires an end Milestone for the project. The last activity should have a successor (i.e., the Milestone) and every activity should be able to trace a link form its successor to this Milestone.
- If the Milestone has float, as Rafael says, **the FF and TF** are equal; the constraint acts as the successor.
- This construct would also mean the critical path as a positive float value.
- Note: This situation should be considered as constraint that project duration is different from EF time of the last activity (calculated through forward pass calculation).
- The critical path in such case is that path which has minimum total float.

# Alternative definitions of FF and TF

## Free Float (Free Slack)

- Free Float (FF) – amount of time a task can be delayed without delaying the *early start* of any immediately following tasks

$$FF_i = ES_{(i+1)} - EF_{(i)} - 1$$

- Free Float is "owned" by *each* task
- It can be used up safely without affecting either the end of the network or any other individual task in the network

## Total Float (Start or Total Slack)

- Total Float (TF) - the amount of time a task can be delayed without delaying the overall project completion time

$$TF = LF - EF \quad or \quad TF = LS - ES$$

- Total float can change as the project progresses and changes are made to the project plan

https://www.youtube.com/watch?v=YboHkzVphzs

# **Further Reading**

Read more about the scheduling network models from:


Jimmie W. Hinze. "Construction Planning and Management," Fourth Edition, 2012, Pearson.

# Thank You

# Questions Please