



Time Planning and Control

Precedence Diagram

■ *Precedence Diagramming*

| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | |
| LS | TF | LF |

- ❑ An important extension to the original activity-on-node concept appeared around 1964.
- ❑ The sole relationship used in *PERT/CPM network* is finish to start type of dependency, with $Fs_{ij} = 0$.
- ❑ Precedence diagramming includes precedence relationships among the activities. In Addition, one may specify a *“lag time”* associated with any of the precedence relationships, which can be used to account for overlapping times among activities.
- ❑ The computation of activity times (published in 1973) is more complex than AON.

■ *Lag / Lead Times*

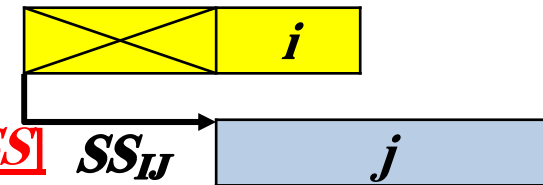
| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | |
| LS | TF | LF |

- In many cases, there is a delay between the completion of one activity and the start of another following or there is a need to show that one activity will *overlap* another in some fashion.
- A successor "lags" a predecessor, but a predecessor "leads" a successor.
- *Lag time can be designated on a dependency line with a positive, negative, or zero value.*
- **Limitations and Disadvantages of Lag:**
 - ☐ Lag would complicate the scheduling process.
 - ☐ Lags are not extensively used except where the time effects are substantial for special project type.

■ Precedence Diagramming Relationships Types and constraint

1. Start-to-Start (SS_{ij})

[(j) cannot start till (i) starts by amount of the SS]

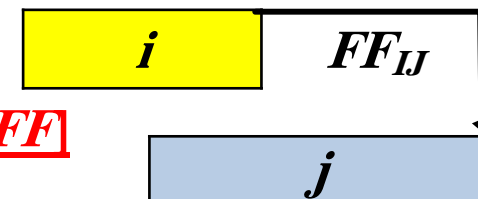


The value of SS_{ij} is equal to the minimum number of time units that must be completed on the preceding activity (i) prior to the start of the successor (j).
“Lag” is always applied to SS relation.

Example $SS_{ij}=3$ [The start of (j) must lag 3 units after the start of (i)]

2. Finish-to-Finish (FF_{ij})

[(j) cannot finish till (i) finishes by amount of the FF]



FF_{ij} is equal to the minimum number of time units that must remain to be completed on the successor (j) after the completion of the predecessor (i).

Example $FF_{ij}=5$ [The finish of (j) must lag 5 units after the finish of (i)]

■ Precedence Diagramming Relationships Types and constraint

3. Finish-to-Start (FS_{ij})



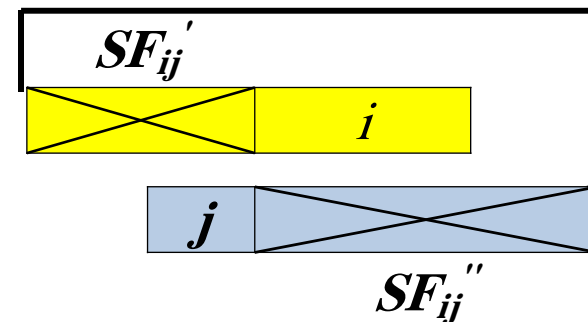
[(j) cannot start till (i) finishes by amount of the FS]

FS_{ij} is equal to the minimum number of time units that must transpire from the completion of the predecessor (i) prior to the start of the successor (j).

Example $FS_{ij}=6$ [The start of (j) must lag 6 units after the finish of (i)]

4. Start-to-Finish (SF_{ij})

[(j) cannot finish till (i) starts (rare)]



SF_{ij} is equal to the minimum number of time units that must transpire from the start of the predecessor (i) to the completion of the successor (j).

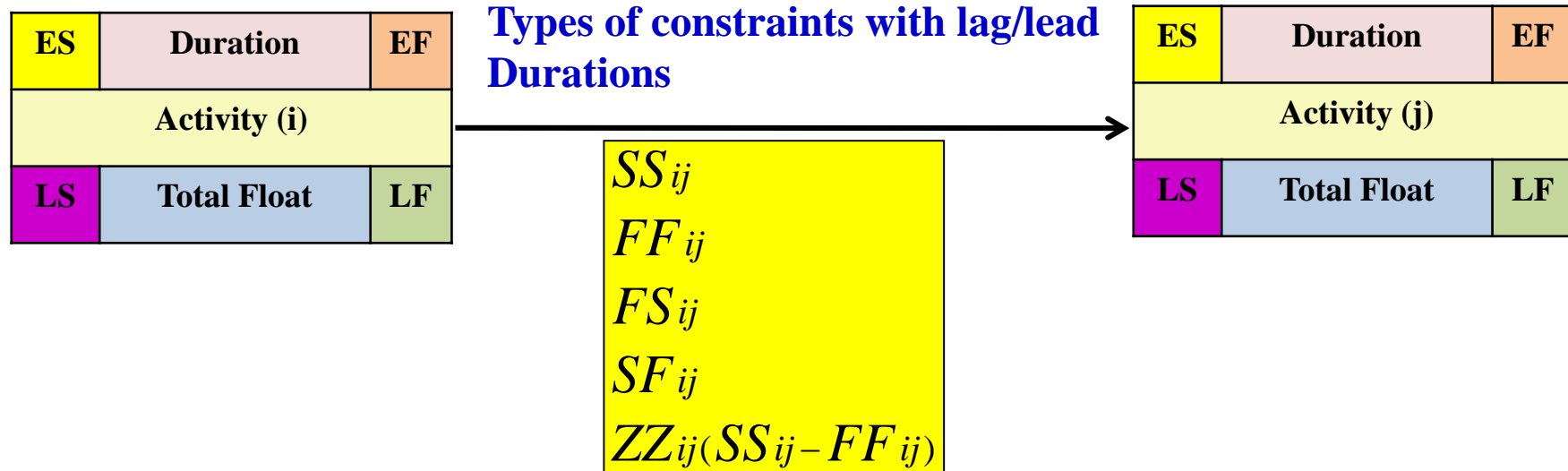
Example $SF_{ij} (SF_{ij}' + SF_{ij}'') = (4+6) = 10$ [The finish of (j) must lag 10 units after the start of (i)]

■ *Precedence Diagramming Relationships Types and constraint*

5. *Start-to-Start and Finish-to-Finish (ZZ_{ij}):*

ZZ_{ij} is a combination of two constraints, i.e., a start-to-start and finish-to-finish relationship. It is written with the SS_{ij} time units first, followed by the FF_{ij} time units.

Example $ZZ_{ij}=5, 6$ [The start of (j) must lag 5 units after the start of (i) ($SS_{ij} = 5$) & The finish of (j) must lag 6 units after the finish of (i) ($FF_{ij} = 6$)]



■ *Precedence Diagram Computation*

| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | FF |
| LS | TF | LF |

Forward Pass Computations

$$[1] ES_j = \text{Max. all } i \left\{ \begin{array}{l} \text{Initial Time} \\ EF_i + FS_{ij} \\ ES_i + SS_{ij} \\ EF_i + FF_{ij} - D_j \\ ES_i + SF_{ij} - D_j \end{array} \right.$$

$$[2] EF_j = ES_j + D_j$$

■ *Precedence Diagram Computation*

| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | FF |
| LS | TF | LF |

Backward Pass Computations

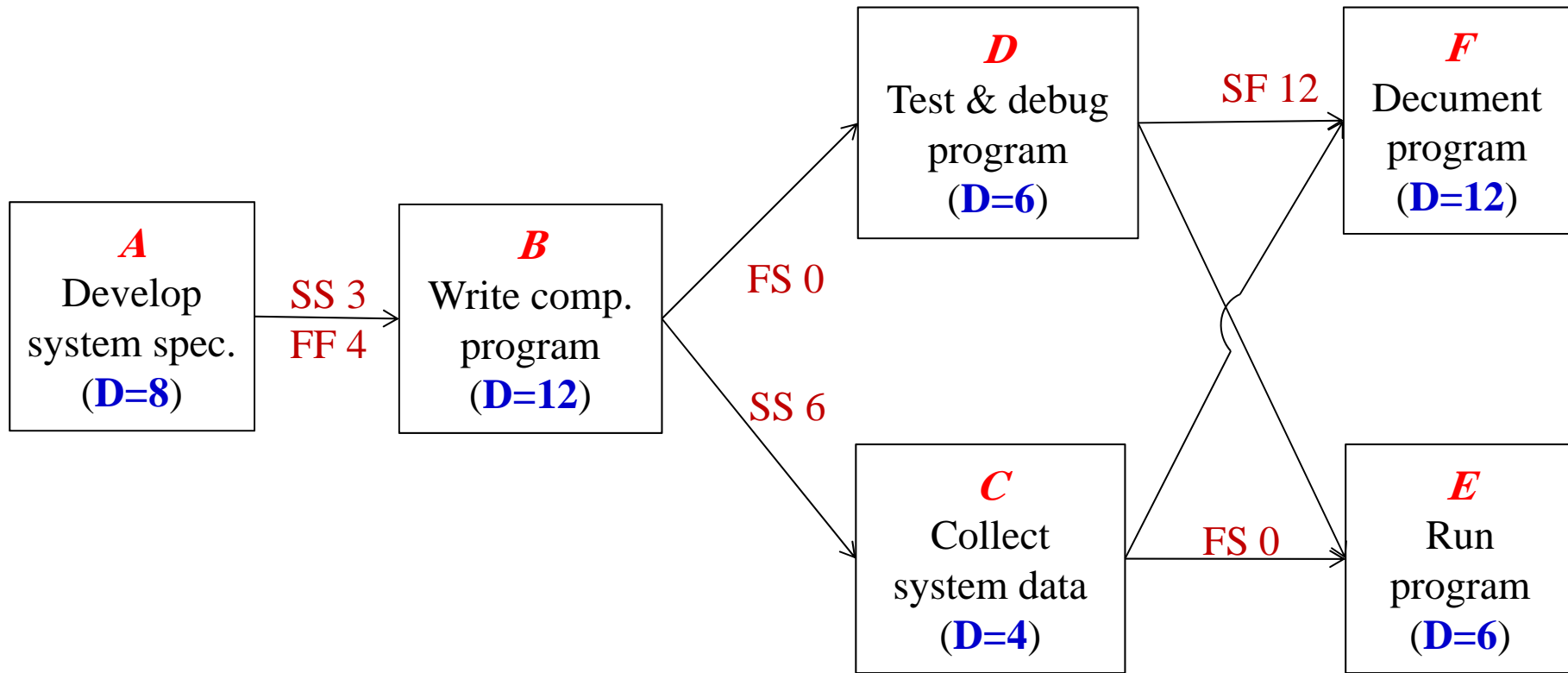
$$[3] \quad LF_i = \text{Min. all } j \quad \left\{ \begin{array}{l} \text{Terminal Time} \\ LS_j - FS_{ij} \\ LF_j - FF_{ij} \\ LS_j - SS_{ij} + D_i \\ LF_j - SF_{ij} + D_i \end{array} \right.$$

$$[4] \quad LS_i = LF_i - D_i$$

Precedence Diagramming Calculations

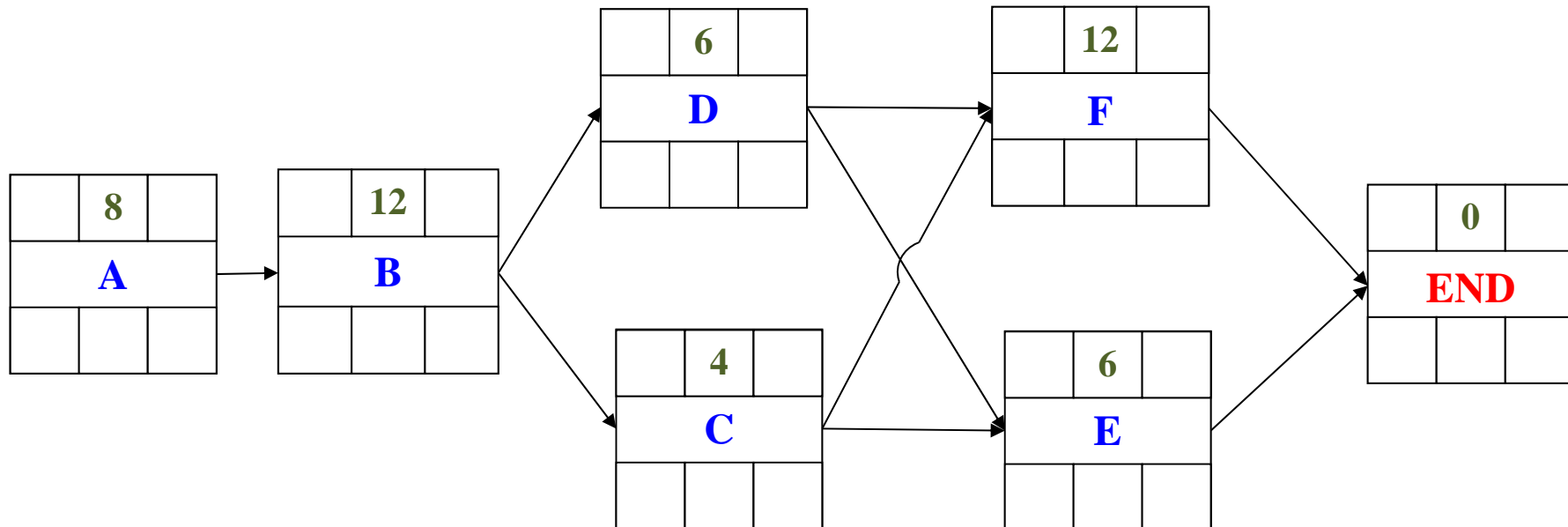
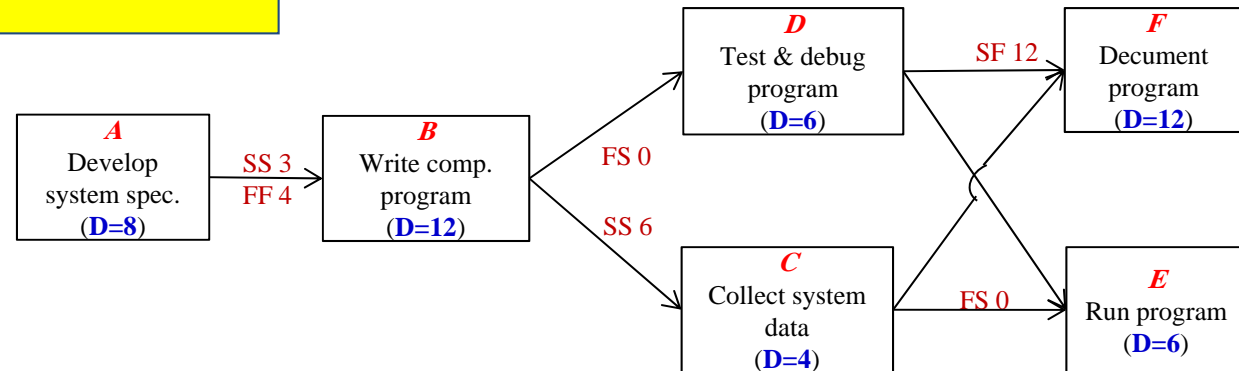
■ **EXAMPLE**

For the given precedence diagram, complete the forward and backward pass calculations. Assume the project starts at T=0, and no splitting on activities is allowed. Also assume that the project latest allowable completion time (project duration) is scheduled for 30 working days.



Precedence Diagramming Calculations

■ AON diagram



Precedence Diagramming Calculations

Example Computation

Activity A

$$ES_A = \text{Initial Time} = 0$$

$$EF_A = ES_A + D_A = 0 + 8 = 8$$

Activity B

$$ES_B = \text{Max}(A) \left\{ \begin{array}{l} \text{Initial Time} = 0 \\ ES_A + SS_{AB} = 0 + 3 = 3 \\ EF_A + FF_{AB} - D_B = 8 + 4 - 12 = 0 \end{array} \right\}$$

$$EF_B = ES_B + D_B = 3 + 12 = 15$$

Activity C

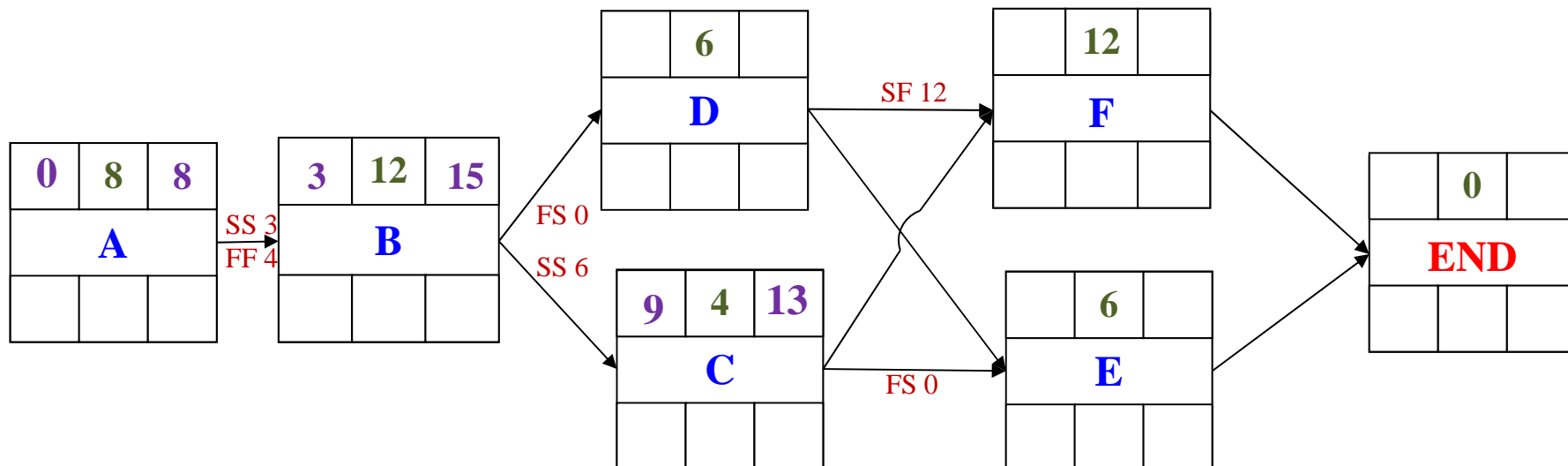
$$ES_C = \text{Max}(B) \left\{ \begin{array}{l} \text{Initial Time} = 0 \\ ES_B + SS_{BC} = 3 + 6 = 9 \end{array} \right\}$$

$$EF_C = ES_C + D_C = 9 + 4 = 13$$

Forward Pass Computations

$$[1] ES_j = \text{Max}(all_i) \left\{ \begin{array}{l} \text{Initial Time} \\ EF_i + FS_{ij} \\ ES_i + SS_{ij} \\ EF_i + FF_{ij} - D_j \\ ES_i + SF_{ij} - D_j \end{array} \right\}$$

$$[2] EF_j = ES_j + D_j$$



Precedence Diagramming Calculations

Example Computation

Activity D

$$ES_D = \text{Max}(B) \left\{ \begin{array}{l} \text{Initial Time} = 0 \\ EF_B + FS_{BD} = 15 + 0 = 15 \end{array} \right\}$$

$$EF_D = ES_D + D_D = 15 + 6 = 21$$

Activity E

$$ES_E = \text{Max}(C, D) \left\{ \begin{array}{l} \text{Initial Time} = 0 \\ EF_D + FS_{DE} = 21 + 0 = 21 \\ \text{OR, } EF_C + FS_{CE} = 13 + 0 = 13 \end{array} \right\}$$

$$EF_E = ES_E + D_E = 21 + 6 = 27$$

Activity F

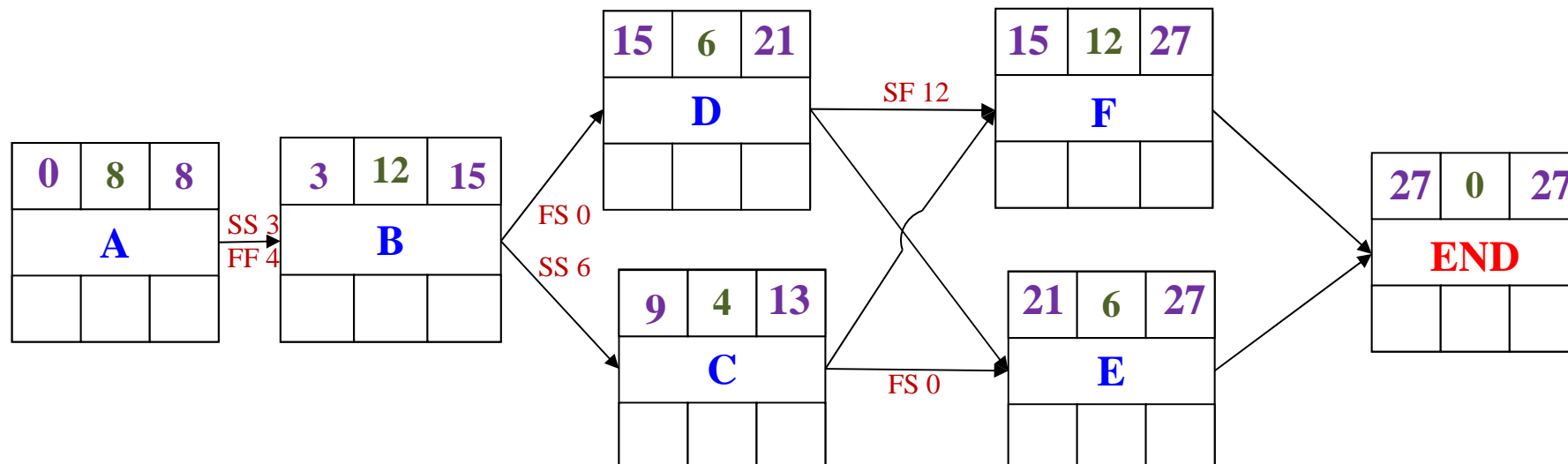
$$ES_F = \text{Max}(D) \left\{ \begin{array}{l} \text{Initial Time} = 0 \\ ES_D + SF_{DF} - D_F = 15 + 12 - 12 = 15 \\ \text{OR, } EF_C + FS_{CF} = 13 + 0 = 13 \end{array} \right\}$$

$$EF_F = ES_F + D_F = 15 + 12 = 27$$

Forward Pass Computations

$$[1] ES_j = \text{Max}(all_i) \left\{ \begin{array}{l} \text{Initial Time} \\ EF_i + FS_{ij} \\ ES_i + SS_{ij} \\ EF_i + FF_{ij} - D_j \\ ES_i + SF_{ij} - D_j \end{array} \right\}$$

$$[2] EF_j = ES_j + D_j$$



Precedence Diagramming Calculations

Example Computation

Activity F

$$LF_F = \text{Terminal Time} = 30$$

$$LS_F = LF_F - D_F = 30 - 12 = 18$$

Activity E

$$LF_E = \text{Terminal Time} = 30$$

$$LS_E = LF_E - D_E = 30 - 6 = 24$$

Activity D

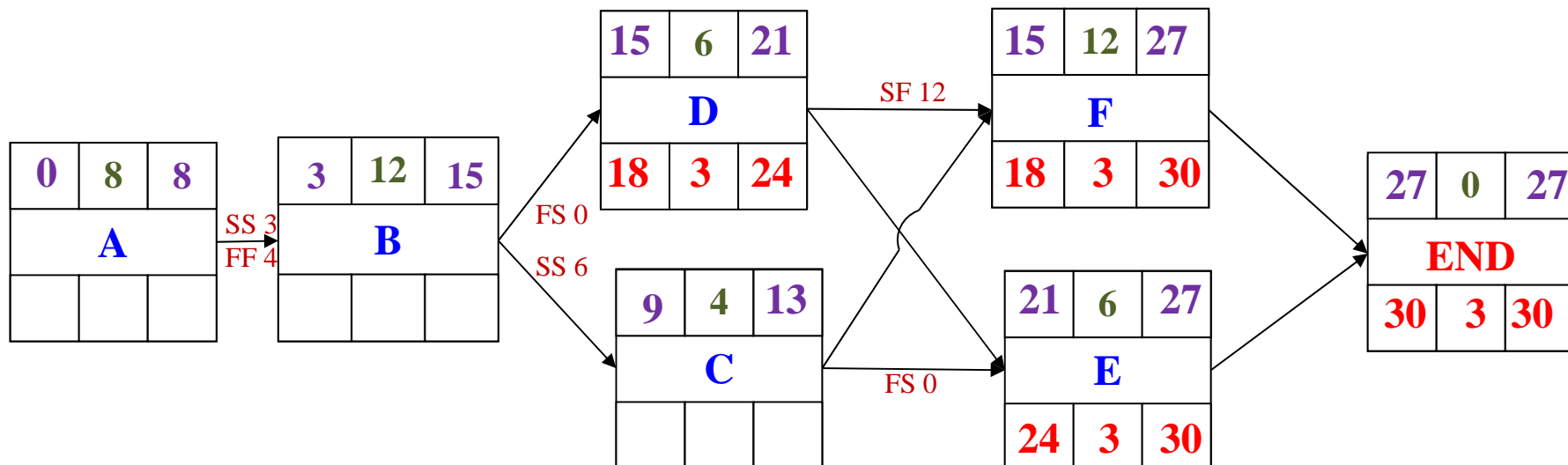
$$LF_D = \text{Min}(E, F) \left\{ \begin{array}{l} \text{Terminal Time} = 30 \\ LS_E - FS_{DE} = 24 - 0 = 27 \\ \text{OR} \\ LF_F - SF_{DF} + D_D = 30 - 12 + 6 = 24 \end{array} \right\}$$

$$LS_D = LF_D - D_D = 24 - 6 = 18$$

Backward Pass Computations

$$[3] LF_i = \text{Min}(all_j) \left\{ \begin{array}{l} \text{Terminal Time} \\ LS_j - FS_{ij} \\ LF_j - FF_{ij} \\ LS_j - SS_{ij} + D_i \\ LF_j - SF_{ij} + D_i \end{array} \right\}$$

$$[4] LS_i = LF_i - D_i$$



Precedence Diagramming Calculations

Example Computation

Activity C

$$LF_C = \min(E) \left\{ \begin{array}{l} \text{Terminal Time} = 30 \\ LS_E - FS_{CE} = 24 - 0 = 24 \\ \text{Or} \\ LS_F - FS_{CF} = 18 - 0 = 18 \end{array} \right.$$

$$LS_C = LF_C - D_C = 18 - 4 = 14$$

Activity B

$$LF_B = \min(C, D) \left\{ \begin{array}{l} \text{Terminal Time} = 30 \\ LS_D - FS_{BD} = 18 - 0 = 18 \\ \text{OR} \\ LS_C - SS_{BC} + D_B = 14 - 6 + 12 = 20 \end{array} \right.$$

$$LS_B = LF_B - D_B = 18 - 12 = 6$$

Activity A

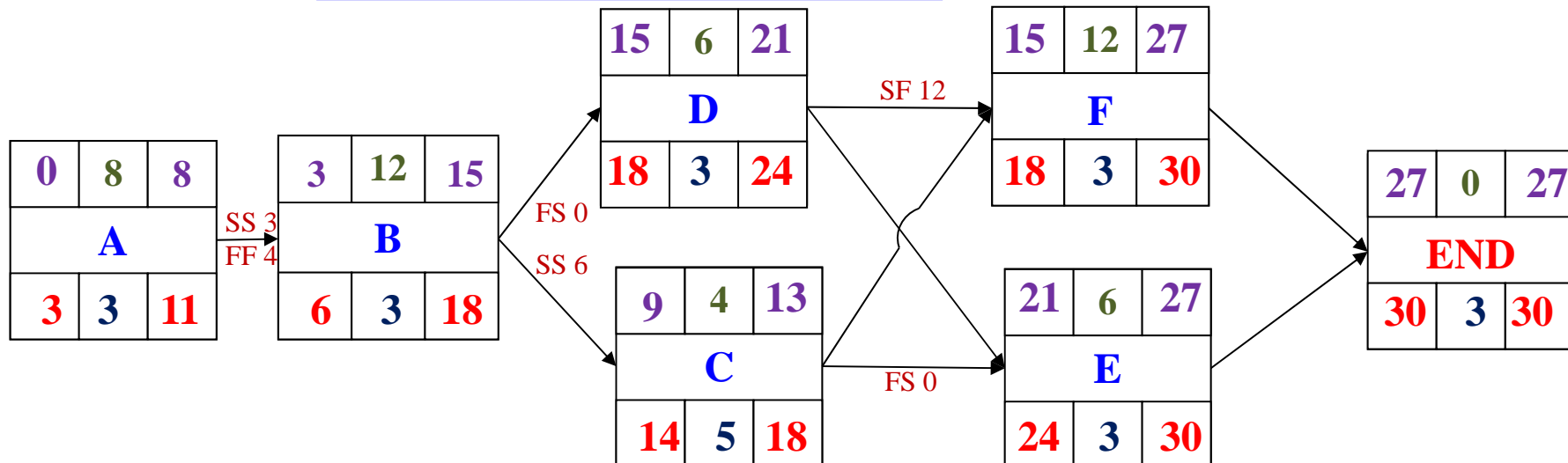
$$LF_A = \min(B) \left\{ \begin{array}{l} \text{Terminal Time} = 30 \\ LF_B - FF_{AB} = 18 - 4 = 14 \\ LS_B - SS_{AB} + D_A = 6 - 3 + 8 = 11 \end{array} \right.$$

$$LS_A = LF_A - D_A = 11 - 8 = 3$$

Backward Pass Computations

$$[3] LF_i = \min(\text{all } j) \left\{ \begin{array}{l} \text{Terminal Time} \\ LS_j - FS_{ij} \\ LF_j - FF_{ij} \\ LS_j - SS_{ij} + D_i \\ LF_j - SF_{ij} + D_i \end{array} \right.$$

$$[4] LS_i = LF_i - D_i$$



Precedence Diagramming Calculations

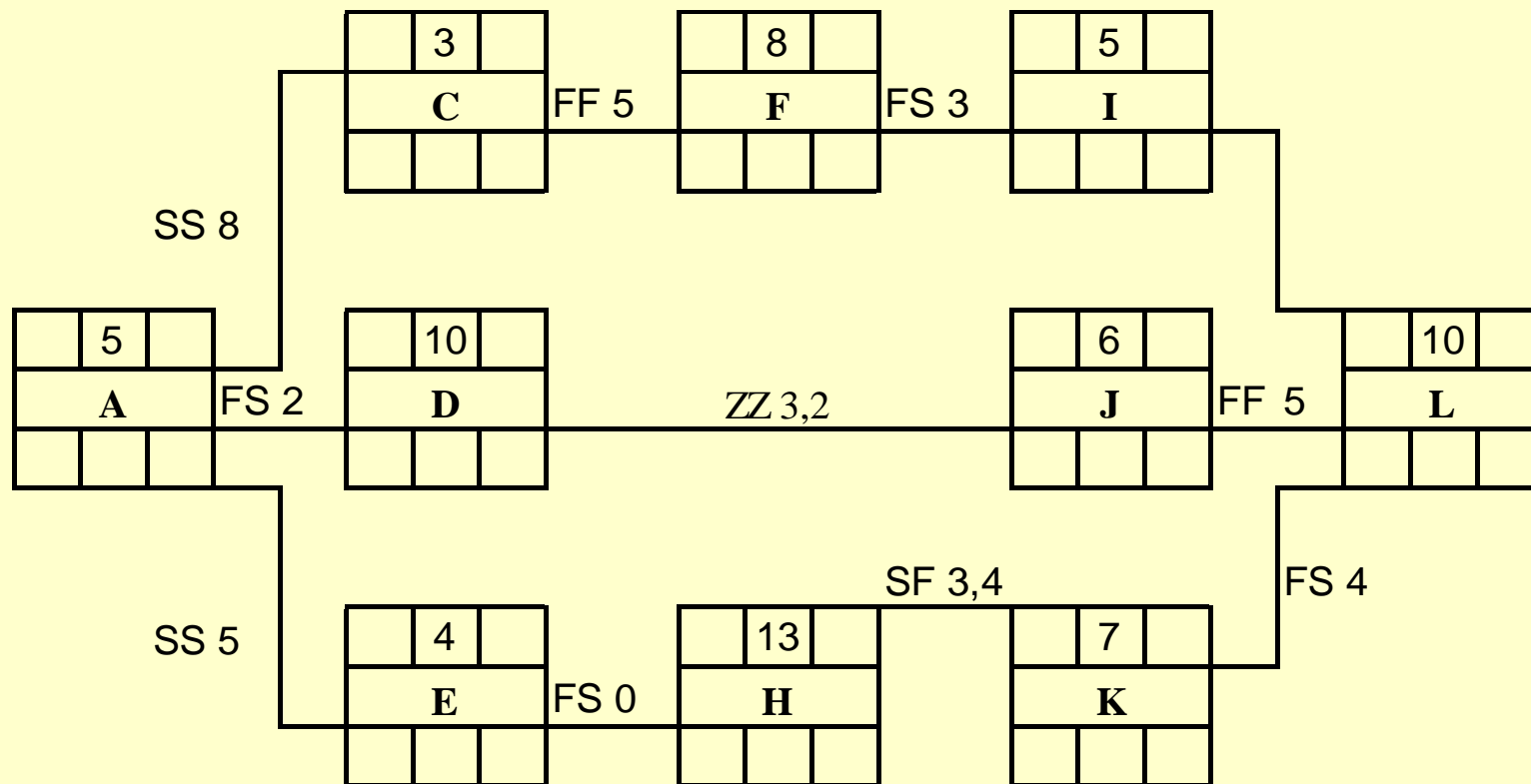
Computing Slack Time (Float Time)

| Activity | Earliest Start ES | Earliest Finish EF | Latest Start LS | Latest Finish LF | Slack LS – ES | On Critical Path |
|----------|-------------------------|--------------------------|-----------------------|------------------------|------------------|------------------------|
| A | 0 | 8 | 3 | 11 | 3 | No |
| B | 3 | 15 | 6 | 18 | 3 | No |
| C | 9 | 13 | 14 | 18 | 5 | No |
| D | 15 | 21 | 15 | 21 | 3 | No |
| E | 21 | 27 | 24 | 30 | 3 | No |
| F | 15 | 27 | 18 | 30 | 3 | No |

Example 2

| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | FF |
| LS | TF | LF |

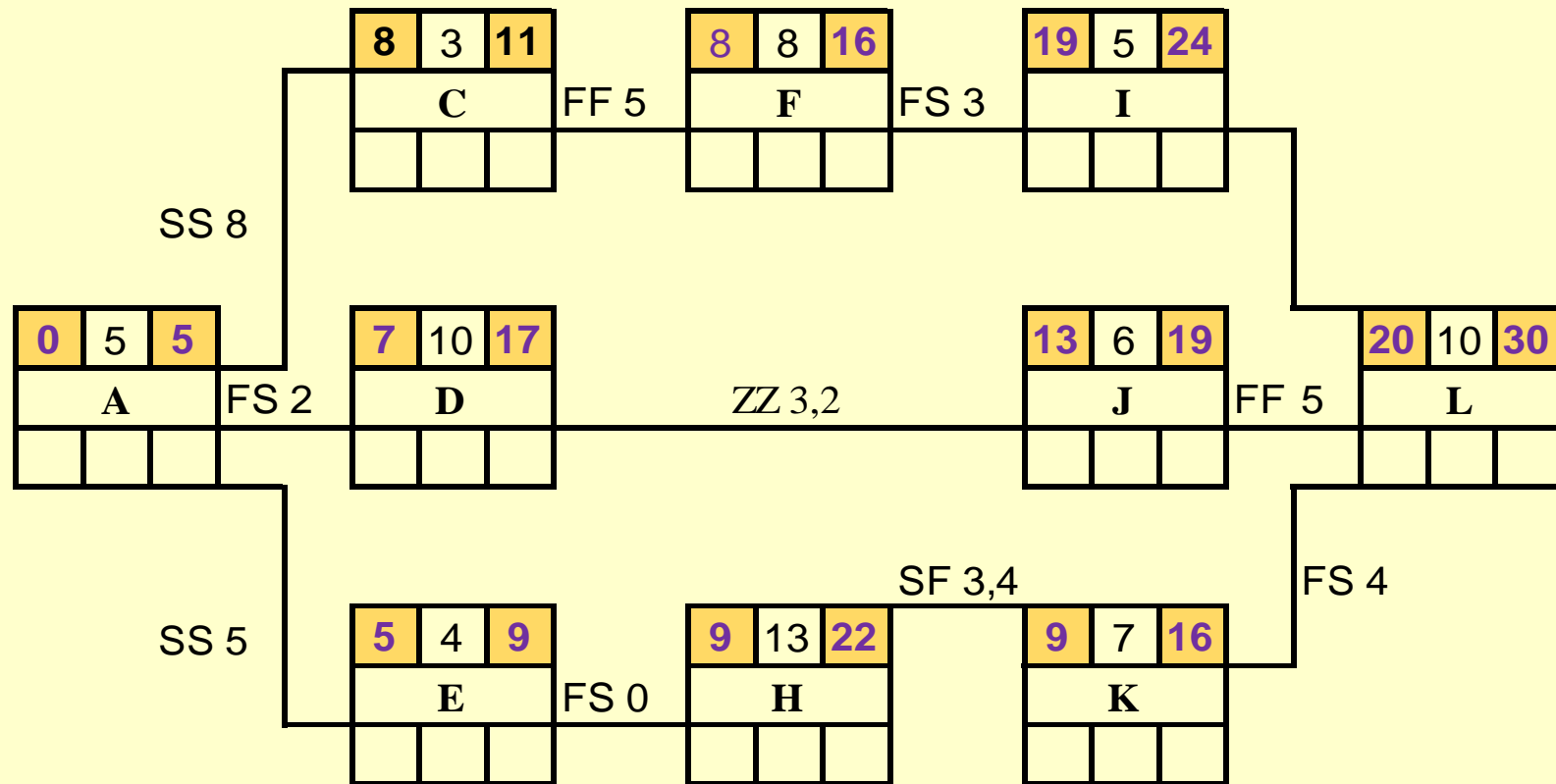
Given the precedence network for a small engineering project with activity durations in working days, it is required to compute the activity times (ES, EF, LS, and LF) and total floats (TF) and then indicate the critical activities.



Example 2

Calculate the Early activity times (ES and EF).

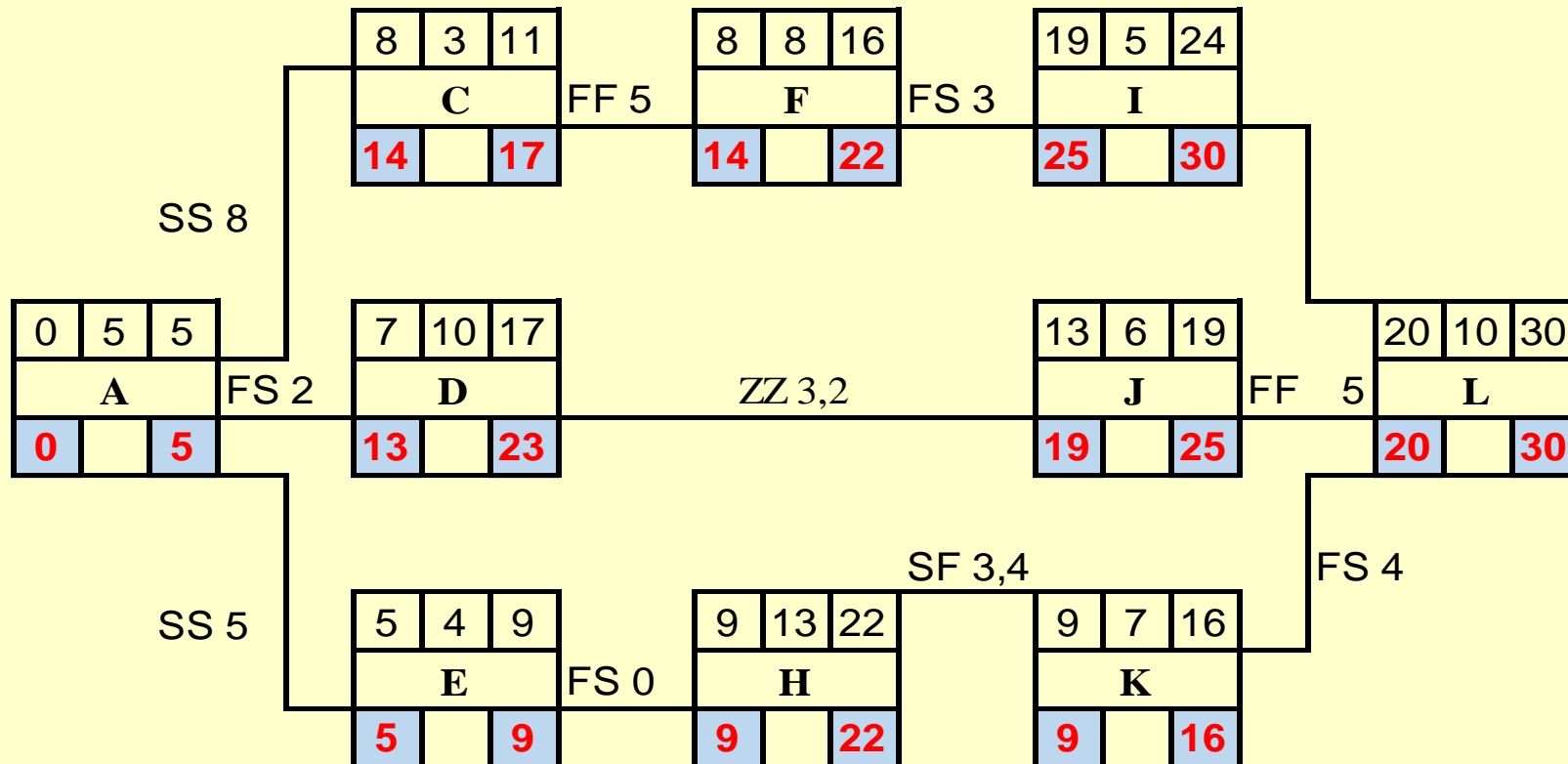
| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | FF |
| LS | TF | LF |



Example 2

Calculate the late activity times (LS and LF).

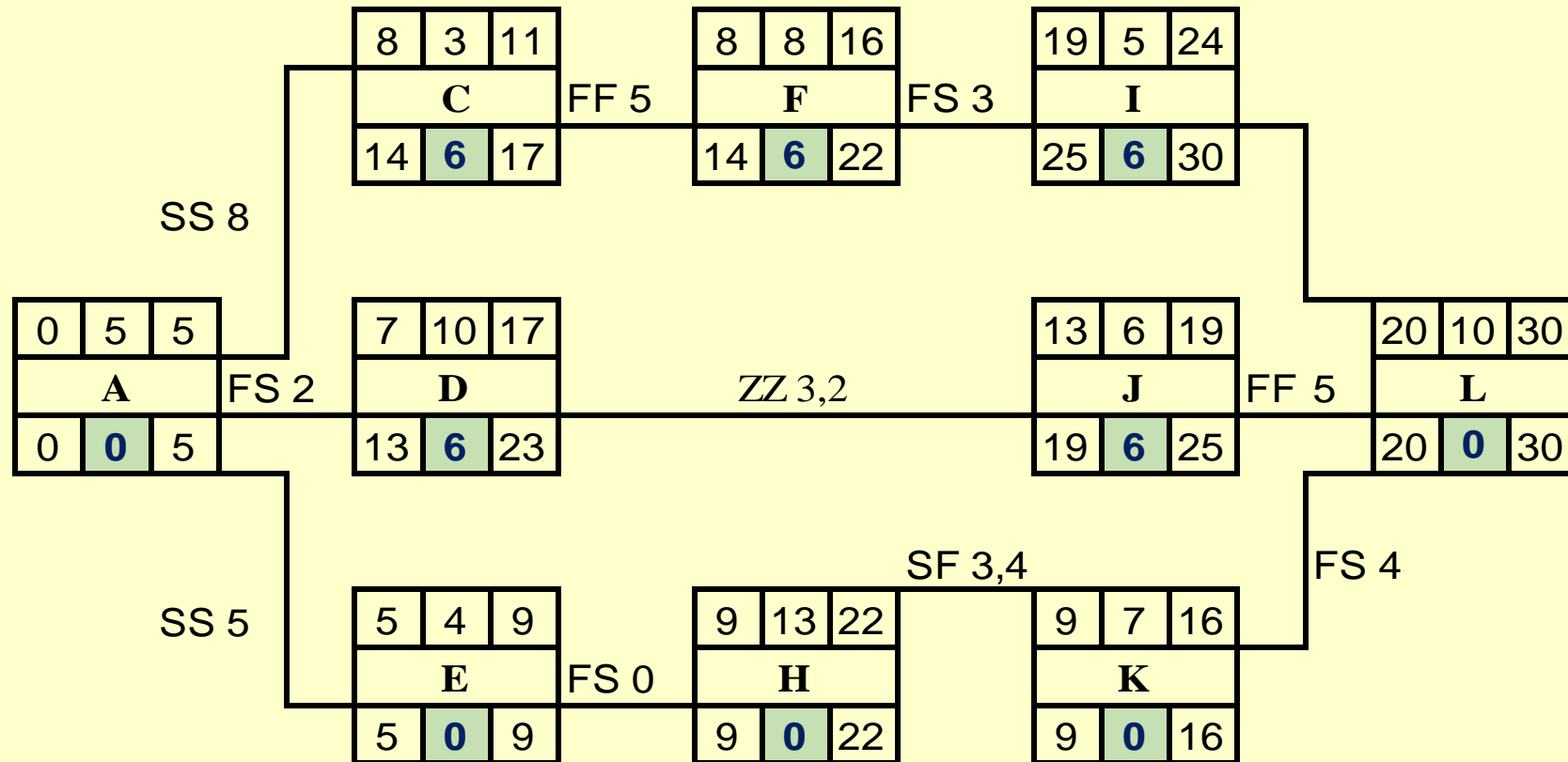
| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | | FF |
| LS | TF | LF |



Example 2

Calculate Total Float for an activity.

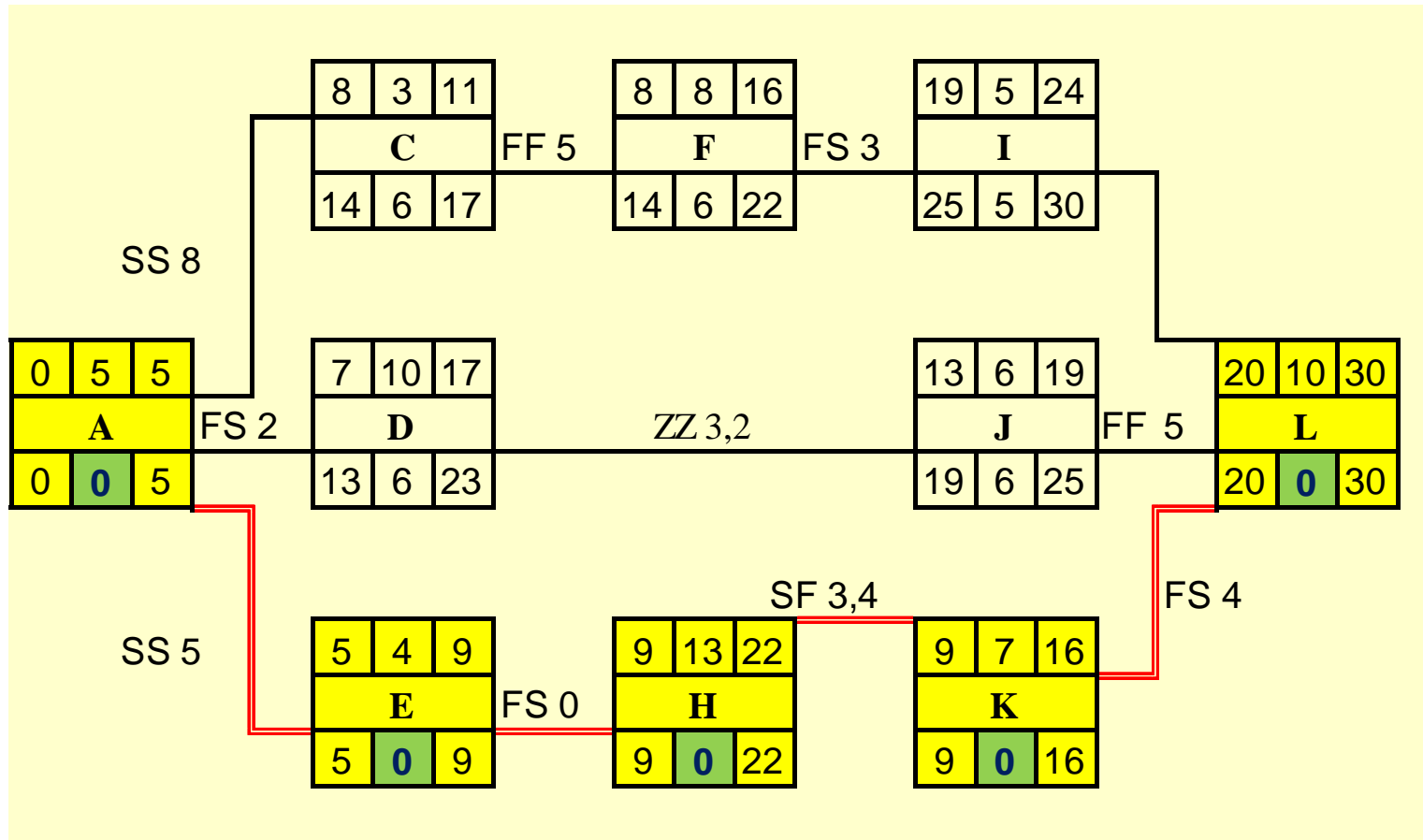
| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | FF | |
| LS | TF | LF |



Example 2

Indicate the critical activities.

| | | |
|-------------|----|----|
| ES | D | EF |
| Activity ID | FF | |
| LS | TF | LF |



Notes on Schedule

■ HAMMOCK ACTIVITY

- An activity that extends from one activity to another, but which has no estimated duration of its own.
- It is time-consuming and requires resources, but its duration is controlled, not by its own nature, but by the two activities between which it spans.
- Its **ES and LS** times are determined by the activity where it begins and its **EF and LF** times are dictated by the activity at its conclusion.
- Examples: Dewatering, Haul road maintenance

Notes on Schedule

■ **MILESTONES**

- Milestones are points in time that have been identified as being important intermediate **reference points** during the accomplishment of the work.
- Milestone events can include dates imposed by the customer for the finishing of certain tasks as well as target dates set by the project manager for the completion of certain segments of the work.

Notes on Schedule

■ **MILESTONES**

- Distinctive **geometric figure** is preferred to represent a milestone (circles, ovals, or other shapes) can be used.
- Any information pertaining to a milestone and considered to be useful may be entered.

Notes on Schedule

Reducing Project Duration

How can you shorten the schedule?

Via

- Reducing scope (or quality)
- Adding resources
- Concurrency (perform tasks in parallel)
- Substitution of activities