2.3.5 Project planning

Definitions:

A *project* consists of a set of m activities with their duration: activity A_i has duration $d_i \geq 0$, $i = 1,\ldots, m$.

estimate

Some pairs of activities are subject to a *precedence constraint*: $A_i \propto A_j$ indicates that A_j can start only after the end of A_i.

Example

Activities: A, B, C, D, E

Precedences: $A \propto B$, $A \propto C$, $B \propto D$, $C \propto D$, $B \propto E$.

Model

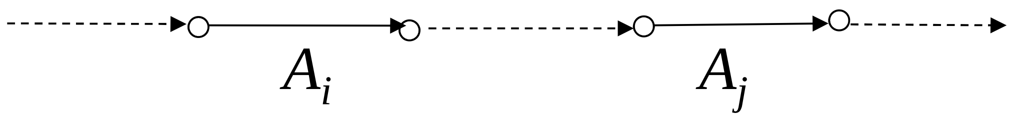
A project can be represented by a directed graph $G = (N, A)$:

arc $\leftrightarrow$ activity

arc length = activity duration.

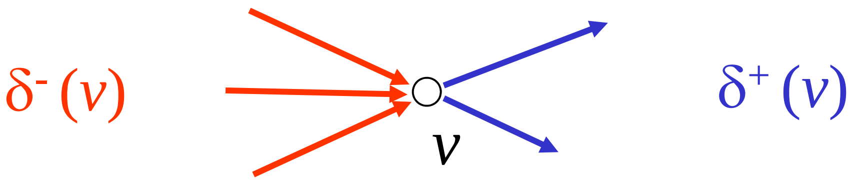
To account for the precedence constraints, the arcs must be positioned so that:

$A_i \propto A_j \iff$ there exists a directed path where the arc associated to A_i preceeds the arc associated to A_j



Therefore we have:

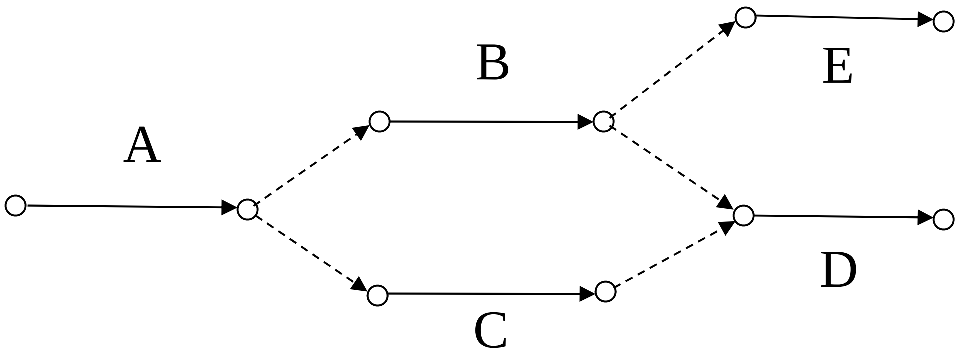
Node v ≈ event corresponding to the end of all the activities $(i, v) \in \delta^-(v)$ and hence to the possible beginning of all those $(v, j) \in \delta^+(v)$.



Example

Activities: A, B, C, D, E

Precedences: $A \propto B$, $A \propto C$, $B \propto D$, $C \propto D$, $B \propto E$



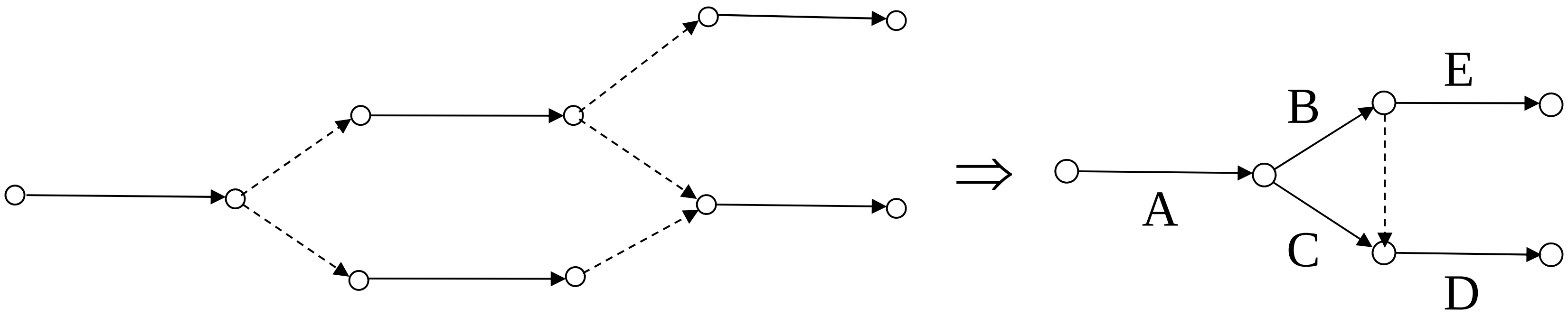
Dummy activities with duration = 0

Property

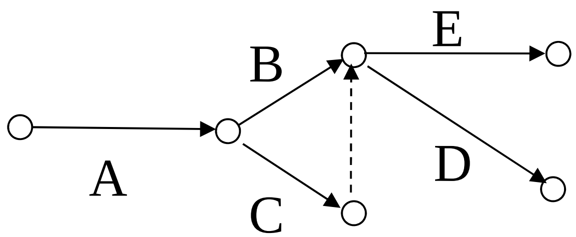
The directed graph G representing any project is <u>acyclic</u> (it is a DAG).

By contradiction: if $A_{i1} \propto A_{12}, \ldots, A_{jk} \propto A_{ki}$ there would be a logical contradiction.

The above graph G can be simplified by contracting some arcs:



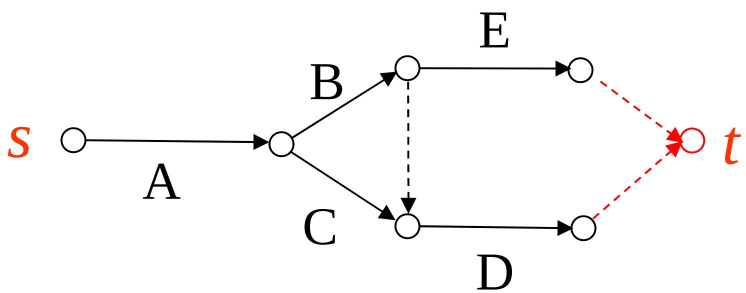
When simplifying the graph, pay attention not to introduce unwanted precedence constraints.



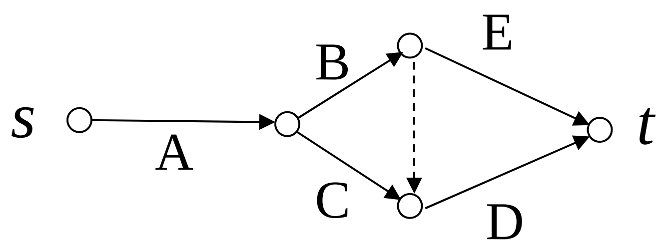
If this dummy activity is maintained, <u>unwanted</u> $C \propto E$ is implied, besides $A \propto B$, $A \propto C$, $B \propto D$, $C \propto D$, $B \propto E$.

Artifical <u>nodes and/or artificial arcs</u> are introduced so that graph G

- contains a unique <u>*initial node s*</u> corresponding to the event "beginning of the project",
- contains a unique <u>*final node t*</u> corresponding to the event "end of the project",
- does not contain multiple arcs (with same endpoints).

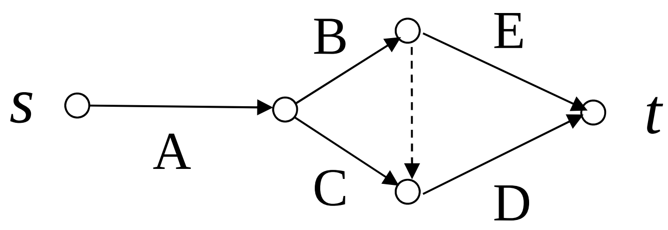


or



Problem

Given a project (set of activities with durations and precedence constraints), schedule the activities so as to minimize the overall project duration , i.e., the time needed to complete all activities.



Property

Minimum overall project duration = length of a longest path from *s* to *t*.

Since any *s*-*t* path represents a sequence of activities that must be executed in the specified order, its length provides a lower bound on the minimum overall project duration.

Critical path method (CPM)

Determines

- a *schedule* (a plan for executing the activities specifying the order and the alotted time) that minimizes the overall project duration,
- the *slack* of each activity, i.e., the amount of time by which its execution can be delayed, without affecting the overall minimum duration of the project.

Initialization: Construct the graph *G* representing the project and determine a topological order of the nodes.

Phase I: Consider the nodes by increasing indices and, for each node $h \in N$, determine:

the <u>*earliest time*</u> Tmin[h] at which the event associated to node h can occur (Tmin[n] corresponds to the <u>*minimum*</u> *overall* <u>*project duration*</u>).

Phase II: Consider the nodes by decreasing indices and, for each node $h \in N$, determine:

the <u>*latest time*</u> Tmax[h] at which the event associated to node h can occur without delaying the completion of the project beyond the <u>minimum project duration</u>.

Phase III: For each activity $(i, j) \in A$, compute the <u>*slack*</u>

$$\sigma_{ij} = \text{Tmax}[j] - \text{Tmin}[i] - d_{ij}.$$

Example

Consider the following project:

Act.	Duration	Predecessors
A	3	-
B	2	A
C	3	A
D	3	C
E	4	B,C
F	3	B
G	1	E,D
H	4	C
I	2	F

Precedence constraints:

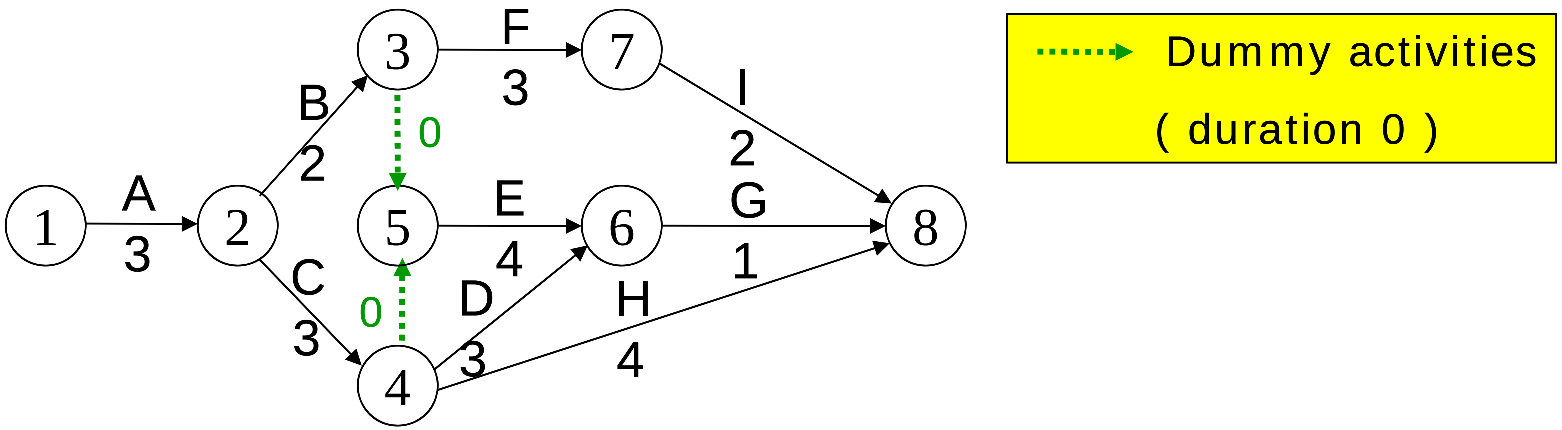
$A \propto B$, $A \propto C$, $C \propto D$, $B \propto E$, $C \propto E$, $B \propto F$, $E \propto G$, $D \propto G$, $C \propto H$, $F \propto I$

Determine the overall minimum duration of the project and the slack for each activity.

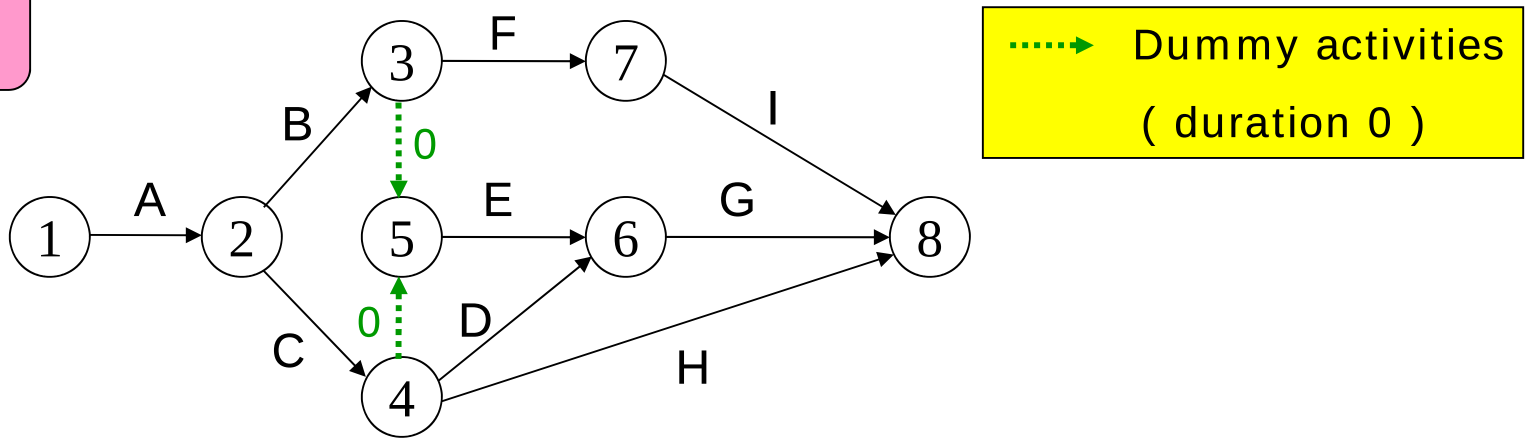
Activities: *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H*, *I*

Precedence constraints: A∝B, A∝C, C∝D, B∝E, C∝E,
B∝F, E∝G, D∝G, C∝H, F∝I

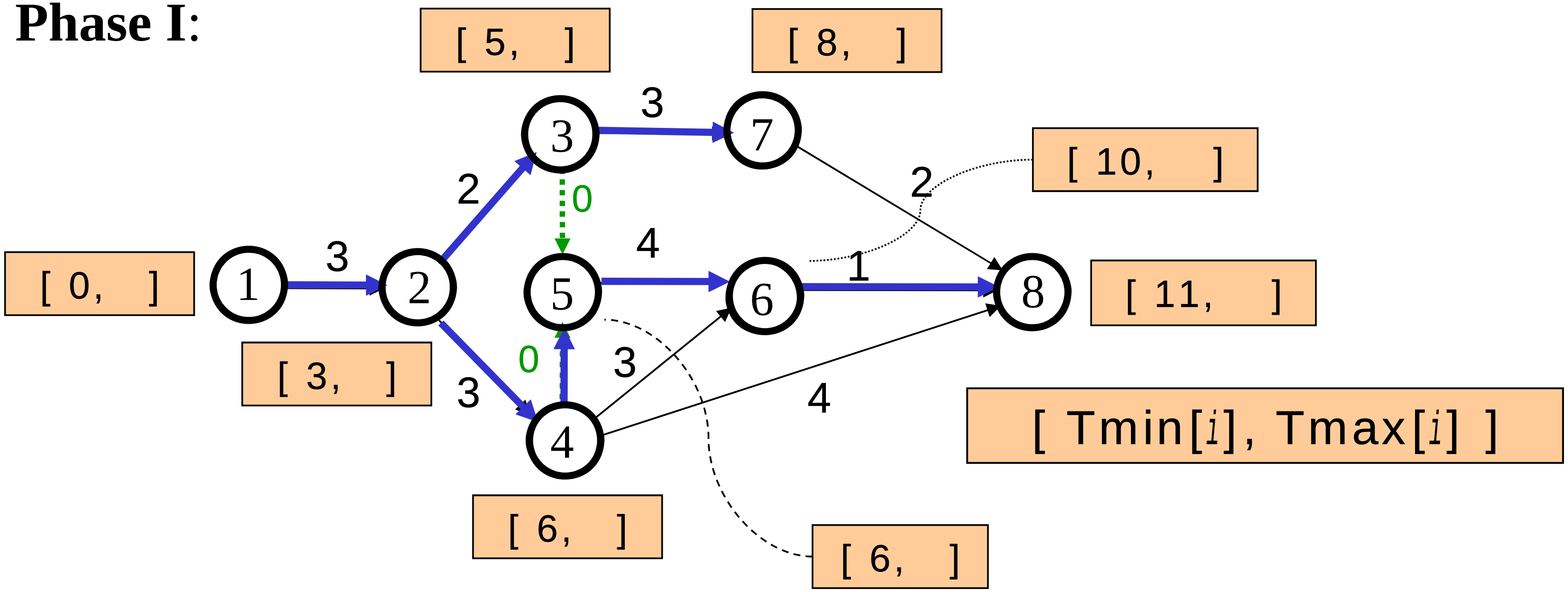
Graphical model:



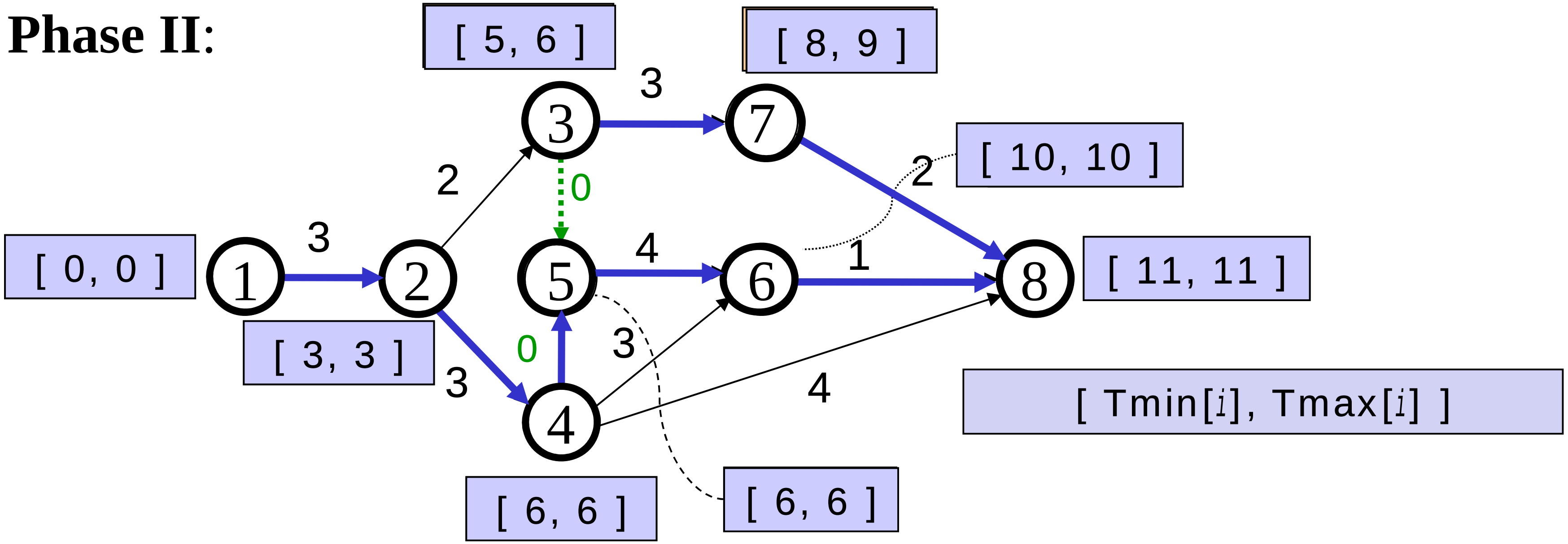
Example

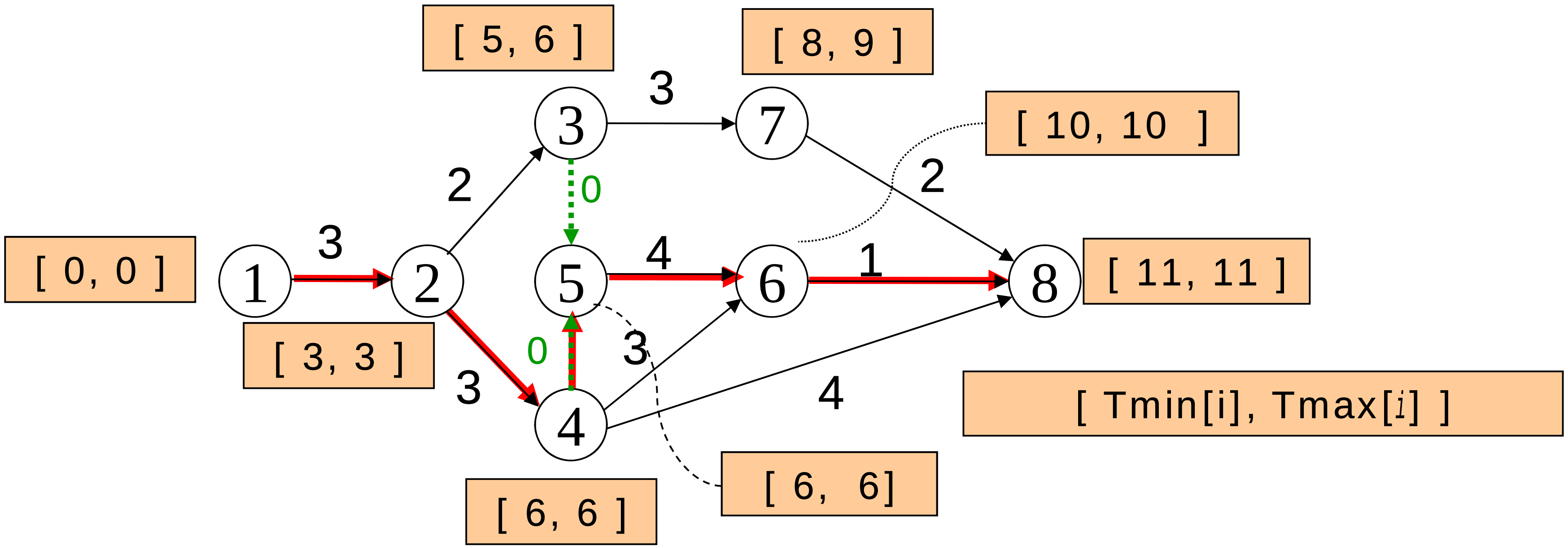


Phase I:



Phase II:





Longest path

Pseudocode of critical path method (CPM)

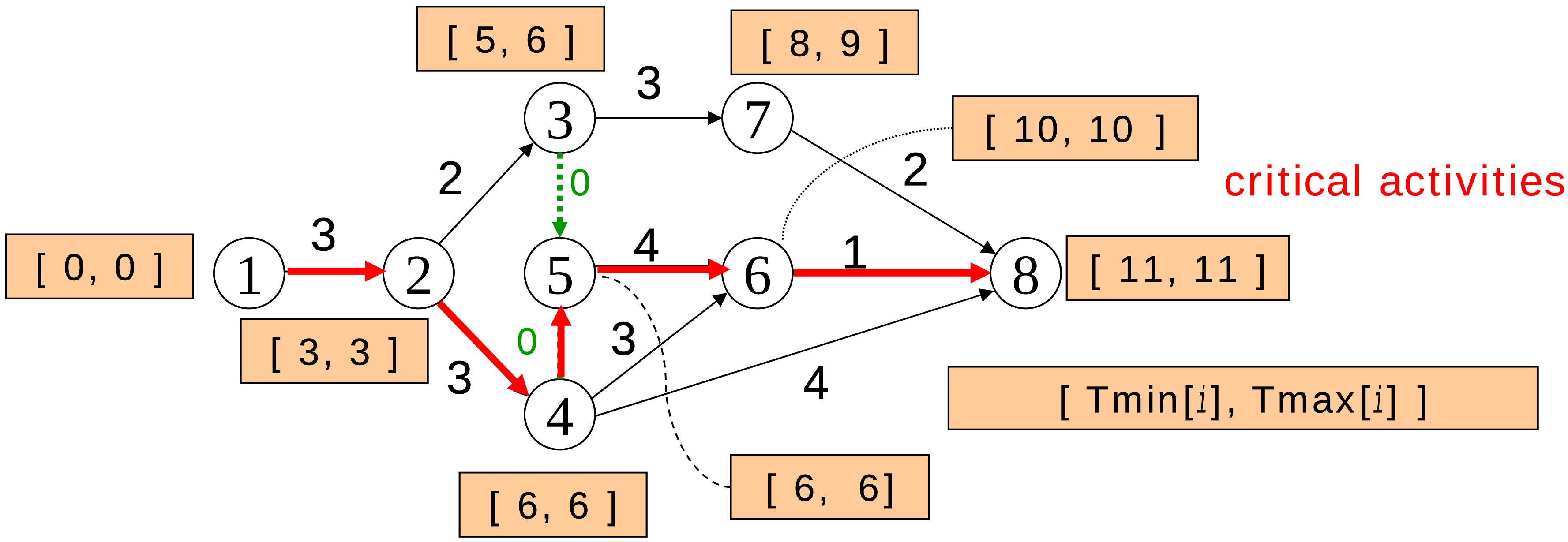
Input: $G = (N, A)$ with $n = |N|$, duration d_{ij} associated to each arc $(i, j) \in A$

Output: Tmin[i] and Tmax[i] for i =1,…, n

```
BEGIN
  Order the nodes topologically;                              O(n+m)
  Tmin[1] := 0;
  FOR h:=2 TO n DO
    Tmin[h] := MAX{ Tmin[i] + d_ih : (i,h) ∈ δ⁻(h) };         O(n+m)
  END-FOR
  Tmax[n] := Tmin[n];  /* minimum project duration */
  FOR h:=n-1 DOWNTO 1 DO
    Tmax[h] := MIN{ Tmax[j] – d_hj : (h,j) ∈ δ⁺(h) };         O(n+m)
  END-FOR
END
```

Overall **complexity**: $O(m)$

Definition: An activity (i, j) with zero slack ($\sigma_{ij} = 0$) is *critical*.

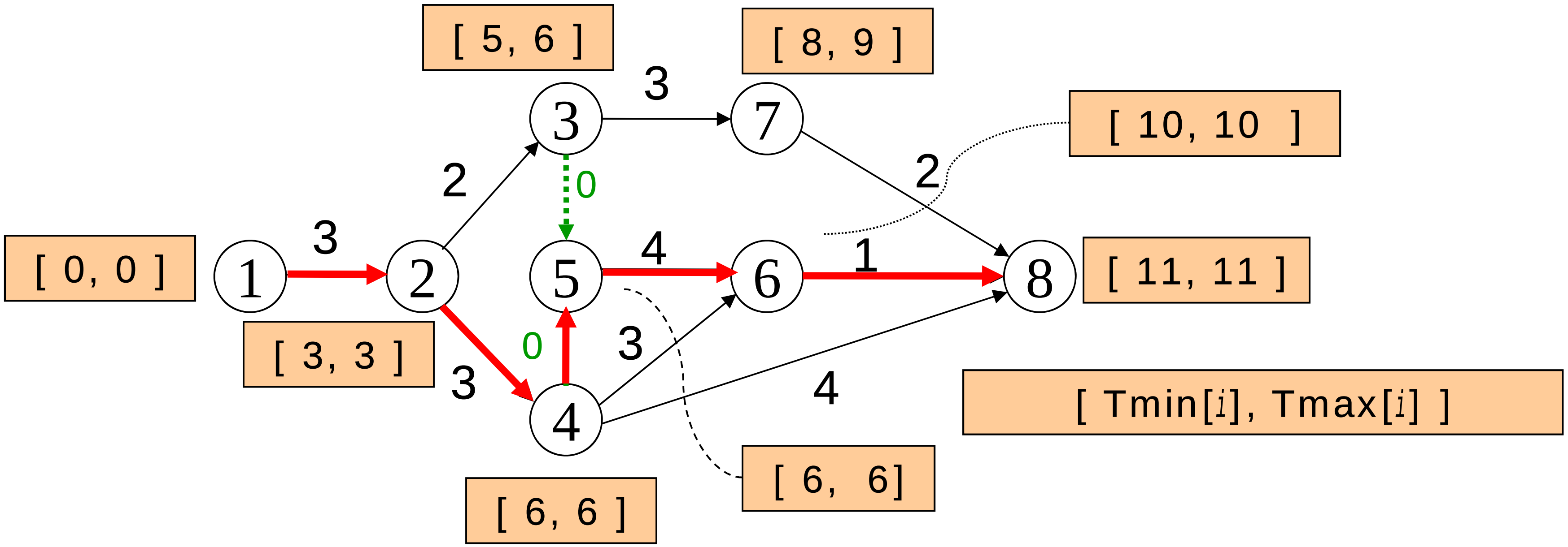


(4, 6) is not crtitical since
σ_{46} = 10 - 6 - 3 =1

σ_{37} = 9 - 5 -3 = 1
σ_{48} = 11 - 6 - 4 = 1

Observation: Tmin[i] = Tmax[i] and Tmin[j] = Tmax[j]
do not suffice to have: σ_{ij} = Tmax[j] - Tmin[i] - d_{ij} = 0 !

Definition: A *critical path* is an $s - t$ path only composed of critical activities (one such path always exists).

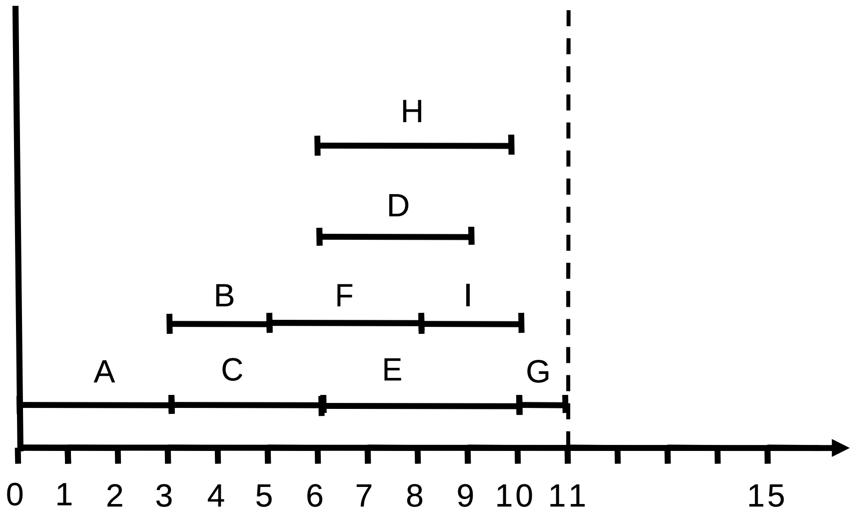


Gantt charts

Introduced in the 1910s by Henry Gantt (1861-1919).

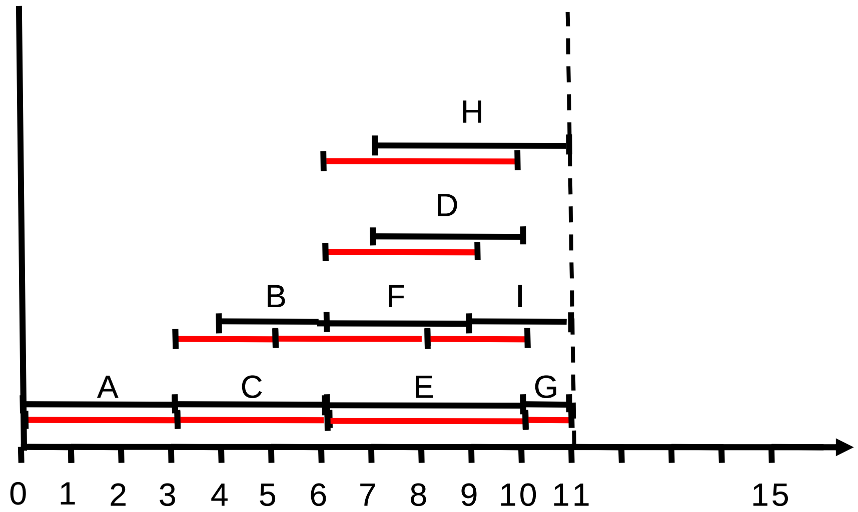
Provide temporal representations of the project.

Gantt chart at earliest : each activity (i, j) starts at time Tmin[i]



(i,j)	d_{ij}	Tmin[i]	Tmax[j]
A	3	0	3
B	2	3	6
C	3	3	6
D	3	6	10
E	4	6	10
F	3	5	9
G	1	10	11
H	4	6	11
I	2	8	11

Gantt chart at latest : each activity (i, j) ends at time Tmax[j]



(i,j)	d_{ij}	Tmin[i]	Tmax[j]
A	3	0	3
B	2	3	6
C	3	3	6
D	3	6	10
E	4	6	10
F	3	5	9
G	1	10	11
H	4	6	11
I	2	8	11