Lesson 4. Weighted Graphs.

1. Definition and Examples.
2. Shortest-Path.
3. Acyclic Graphs. Critical Path Method.
4. Dijkstra's Shortest-Path Algorithm.
5. Floyd-Warshall's Method.
6. Minimum spanning trees.

1. Definitions and Examples.

DEFINITION:

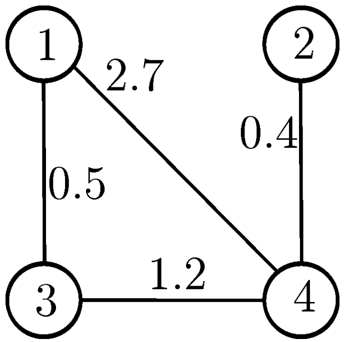
A simple graph G = (V, E) (directed or undirected) is called a **weighted graph** if G has associated a function:

$$W: E \longrightarrow R.$$

This function is called **weighting function.** Now to each edge (arc) e = (v_i, v_j) of this graph, we have assigned a real number called the **weight** of e. This weight is denoted by w_{ij}.

<u>**EXAMPLE:**</u> G=(V, E), V={1,2,3,4}, E={{1,3},{2,4},{3,4},{1,4}}. Consider the function W

$$W: E \longrightarrow R \begin{cases} W(\{1,3\}) = 0.5 \\ W(\{2,4\}) = 0.4 \\ W(\{3,4\}) = 1.2 \\ W(\{1,4\}) = 2.7 \end{cases}$$



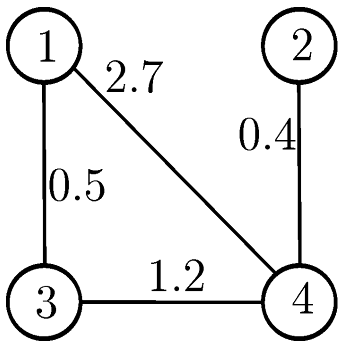
1. Definitions and Examples.

DEFINITION:
Let G = (V, E) be finite weighted graph (directed or undirected) such that $V=\{v_1,v_2,\ldots,v_n\}$. The **weighting matrix** of G is the $n\times n$ matrix defined by:

$$\Omega=\left[a_{ij}\right] / a_{ij}=\begin{cases}\omega_{ij} \text{ if } \{v_i,v_j\}\in E \text{ (if } (v_i,v_j)\in E, \text{ if G directed)}\\ \infty \text{ if } \{v_i,v_j\}\notin E \text{ (if } (v_i,v_j)\notin E, \text{ if G directed)}\end{cases}$$

<u>**EXAMPLE:**</u> G=(V, E), V={1,2,3,4}, E={{1,3},{2,4},{3,4},{1,4}}.

$$W:A\longrightarrow R\begin{cases}W(\{1,3\})=0.5\\ W(\{2,4\})=0.4\\ W(\{3,4\})=1.2\\ W(\{1,4\})=2.7\end{cases}$$

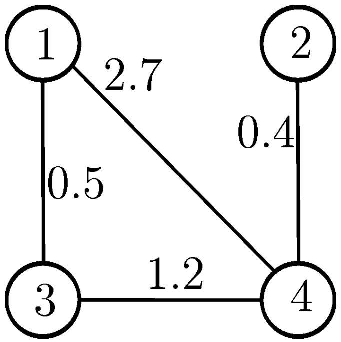


$$\begin{bmatrix}\infty & \infty & 0.5 & 2.7\\ \infty & \infty & \infty & 0.4\\ 0.5 & \infty & \infty & 1.2\\ 2.7 & 0.4 & 1.2 & \infty\end{bmatrix}$$

1. Definitions and Examples.

DEFINITION:
The **weight of a path** in a weighted graph is the sum of the weights of the edges (or arcs) of this path.

EXAMPLE: There are two paths from vertex 1 to 2:



$P_1 \equiv$ 1 3 4 2

$$\omega(C_1) = \omega(\{1,3\}) + \omega(\{3,4\}) + \omega(\{4,2\}) = 0.5 + 1.2 + 0.4 = 2.1$$

$P_2 \equiv$ 1 4 2

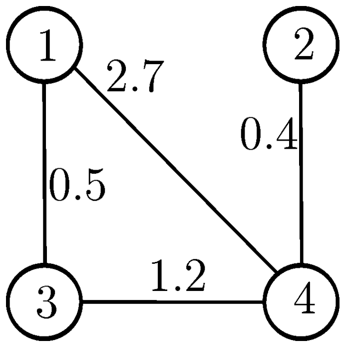
$$\omega(C_2) = \omega(\{1,4\}) + \omega(\{4,2\}) = 2.7 + 0.4 = 3.1$$

1. Definitions and Examples.

DEFINITION:

- In a weighted graph the **shortest path** between two vertices is the the path with **minimum** weight between those vertices.
- In a weighted graph the **longest path** or **critical path** between two vertices is the path with **maximum** weight between those vertices.

EXAMPLE:



$P_1 \equiv$ 1 3 4 2 Shortest path

$$\omega(C_1) = \omega(\{1,3\}) + \omega(\{3,4\}) + \omega(\{4,2\}) = 0.5 + 1.2 + 0.4 = 2.1$$

$P_2 \equiv$ 1 4 2 Longest path – Critical path

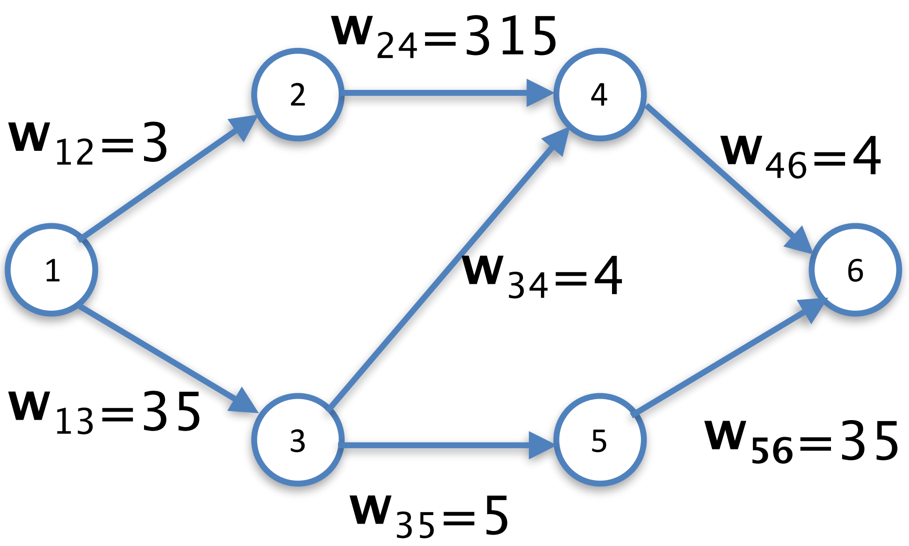
$$\omega(C_2) = \omega(\{1,4\}) + \omega(\{4,2\}) = 2.7 + 0.4 = 3.1$$

2. Shortest-Path.

Notation:

- We assume that the graph is directed and the weights of the arcs are all **nonnegative**.
- We also assume that the vertices of the graph are numbered from 1 to n, so that $\mathbf{w}_{ij}$ represents the weight of the arc (i, j) and the vertex 1 is the origin of the path.
- Furthermore $\mathbf{u}_j$ denotes the weight of shortest path from 1 to j.

Exemple:



3. Acyclic Graphs. Critical Path Method.

THEOREM

A directed graph has no circuits if and only if there is a numbering of the vertices such that if (i, j) is an arc of the graph then $i < j$.

With this numbering, the Bellman's equations can be expressed:

Bellman's equations

$$u_1 = 0$$

$$u_j = \min_{k<j,\ v_k \in \Gamma^{-1}(v_j)} \{u_k + \omega_{kj}\}, \quad j = 2, \ldots, n$$

Numbering Algorithm

Step 1. Initialize $i \leftarrow 1$, $V^{(1)} = V$.

Step 2. Choose $v \in V^{(i)}$ such that $d_{in}(v)=0$ in $G(V_i)$.

Step 3. Number the vertex v as the vertex i.

Set $V^{(i+1)}=V^{(i)} \sim \{v\}$.

Set $i \leftarrow i+1$.

Step 4. If $V^{(i)}=\varnothing$, then STOP.

Otherwise, goto step 2.

3. Acyclic Graphs. Critical Path Method.

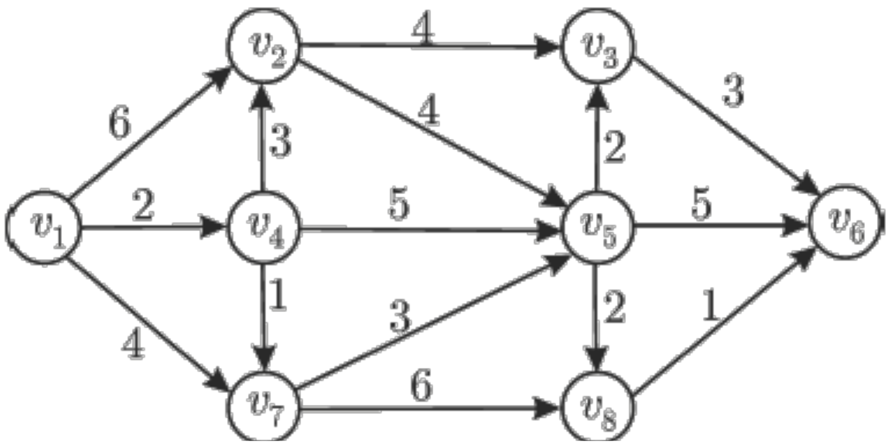
NUMBERING ALGORITHM: EXAMPLE

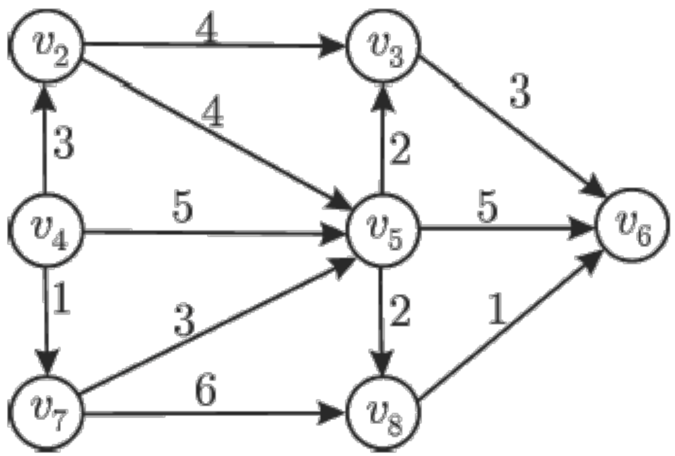
Vertex:	v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8
Number.:	**1**

$i = 1.$

$V^{(1)} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}.$

Choose $v_1 \in V^{(1)}$ / $d_e(v_1) = 0.$





Number v_1 with 1

Remove v_1 from $V^{(1)}$, that is,

$V^{(2)} = V^{(1)} \sim \{v_1\}.$

3. Acyclic Graphs. Critical Path Method.

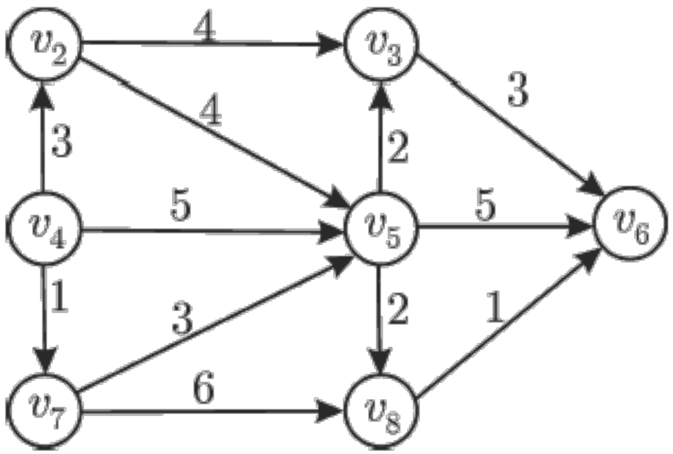
NUMBERING ALGORITHM: EXAMPLE

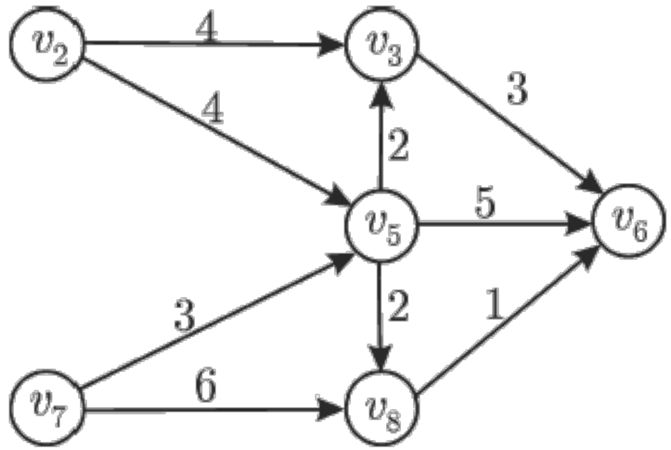
Vertex:	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Number.:	**1**			**2**				

$i = 2.$

$V^{(2)} = \{v_2, v_3, v_4, v_5, v_6, v_7, v_8\}.$

Choose $v_4 \in V^{(2)} \ / \ d_e(v_4) = 0.$





Number v_4 with 2

Remove v_4 from $V^{(2)}$, that is,

$V^{(3)} = V^{(2)} \sim \{v_4\}.$

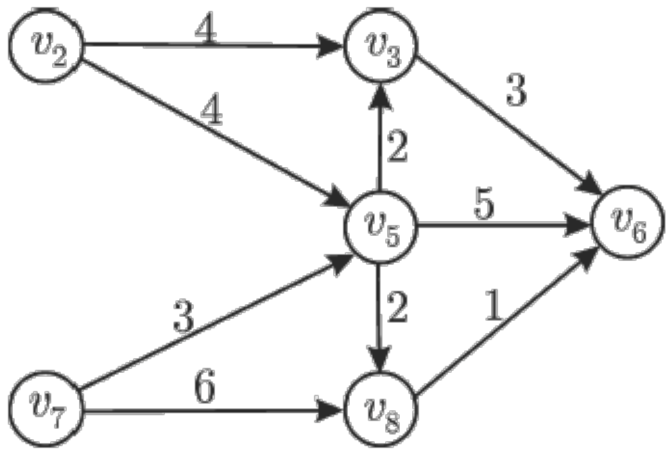
NUMBERING ALGORITHM: EXAMPLE

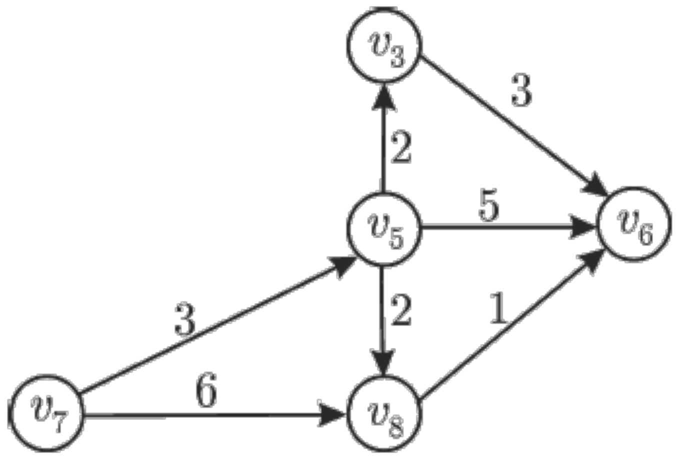
Vertex:	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Number.:	**1**	**3**		**2**				

$i = 3.$

$V^{(3)} = \{v_2, v_3, v_5, v_6, v_7, v_8\}.$

Choose $v_2 \in V^{(3)} \ / \ d_e(v_2) = 0.$





Number v_2 with 3

Remove v_2 from $V^{(3)}$, that is,

$V^{(4)} = V^{(3)} \sim \{v_2\}.$

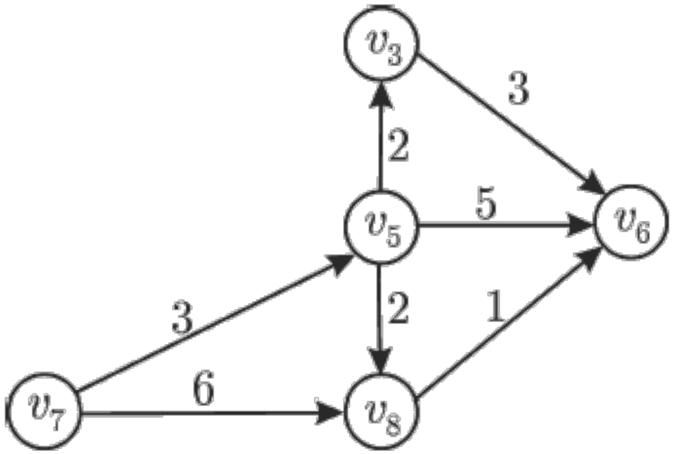
NUMBERING ALGORITHM: EXAMPLE

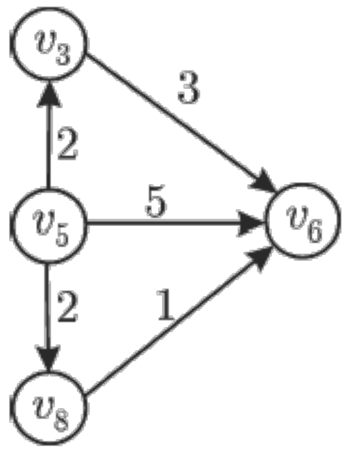
Vertex:	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Number.:	**1**	**3**		**2**			**4**	

$i = 4.$

$V^{(4)} = \{v_3, v_5, v_6, v_7, v_8\}.$

Choose $v_7 \in V^{(4)} \;/\; d_e(v_7) = 0.$





Number v_7 with 4

Remove v_7 from $V^{(4)}$, that is,

$V^{(5)} = V^{(4)} \sim \{v_7\}.$

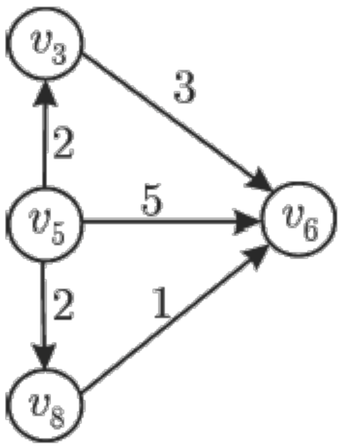
NUMBERING ALGORITHM: EXAMPLE

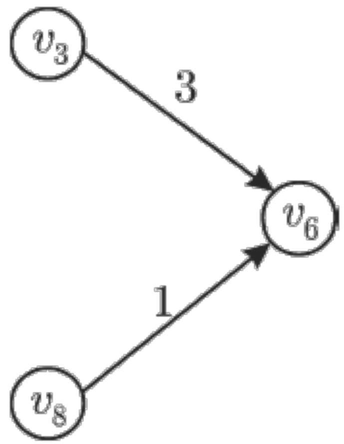
Vertex:	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Number.:	**1**	**3**		**2**	**5**		**4**	

$i = 5.$

$V^{(5)} = \{v_3, v_5, v_6, v_8\}.$

Choose $v_5 \in V^{(5)}$ / $d_e(v_5) = 0.$





Number v_5 with 5

Remove v_5 from $V^{(5)}$, that is,

$V^{(6)} = V^{(5)} \sim \{v_5\}.$

3. Acyclic Graphs. Critical Path Method.

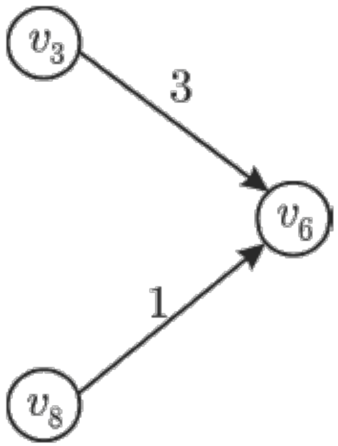
NUMBERING ALGORITHM: EXAMPLE

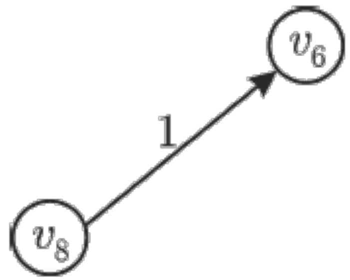
Vertex:	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
Number.:	**1**	**3**	**6**	**2**	**5**		**4**	

$i = 6.$

$V^{(6)} = \{v_3, v_6, v_8\}.$

Choose $v_3 \in V^{(6)} \;/\; d_e(v_3) = 0.$





Number v_3 with 6

Remove v_3 from $V^{(6)}$, that is,

$V^{(7)} = V^{(6)} \sim \{v_3\}.$

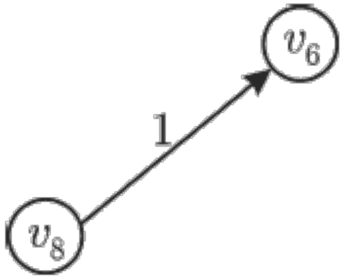
NUMBERING ALGORITHM: EXAMPLE

Vertex:	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
Number.:	**1**	**3**	**6**	**2**	**5**		**4**	**7**

$i = 7.$

$V^{(7)} = \{v_6, v_8\}.$

Choose $v_8 \in V^{(7)} \; / \; d_e(v_8) = 0.$



Number v_8 with 7

Remove v_8 from $V^{(7)}$, that is,

$V^{(8)} = V^{(7)} \sim \{v_8\}.$

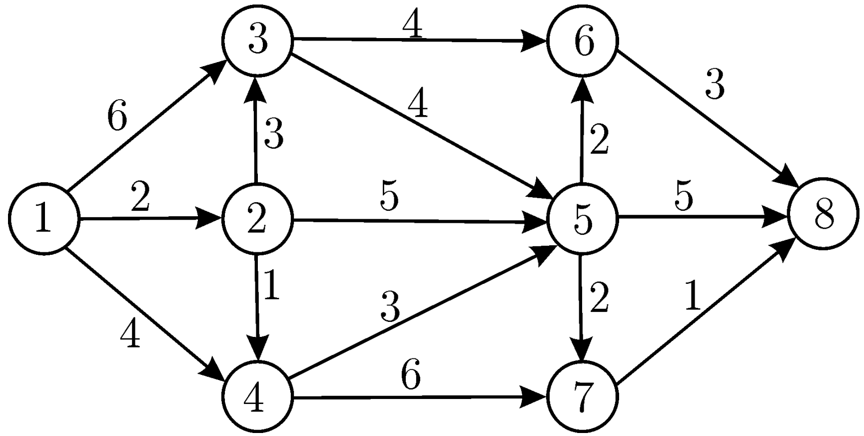
NUMBERING ALGORITHM: EXAMPLE

Vertex:	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Number.:	**1**	**3**	**6**	**2**	**5**	**8**	**4**	**7**

$i = 8.$

$V^{(8)} = \{v_6\}.$

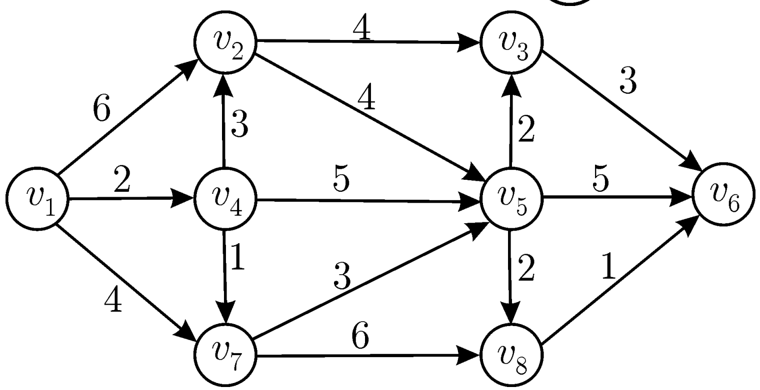
Choose $v_6 \in V^{(8)} \;/\; d_e(v_6) = 0.$



Number v_6 with 8

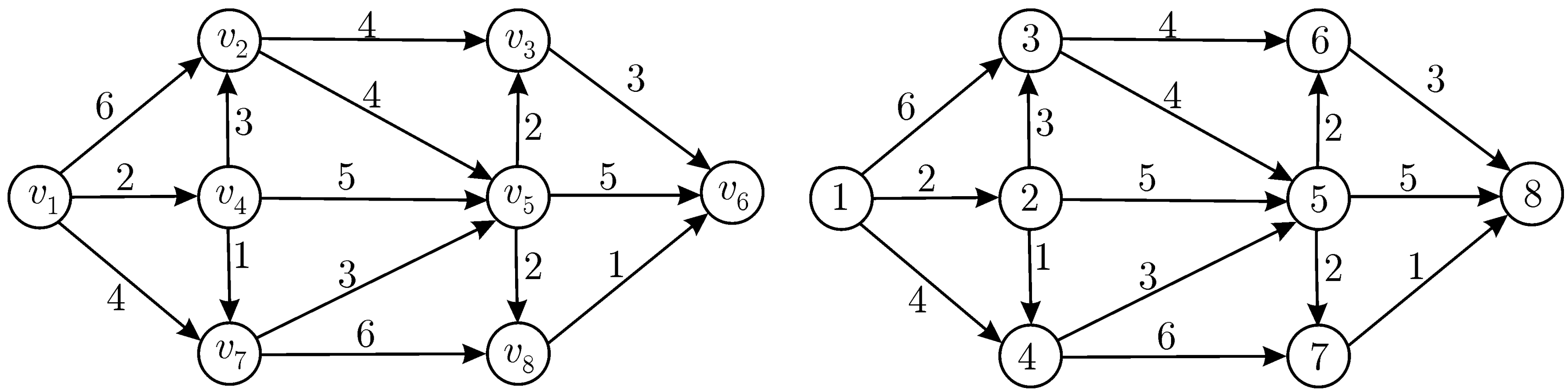
Remove v_6 from $V^{(8)}$, that is,

$V^{(9)} = V^{(8)} \sim \{v_6\} = \emptyset.$



EXAMPLE:

Consider the directed graph:

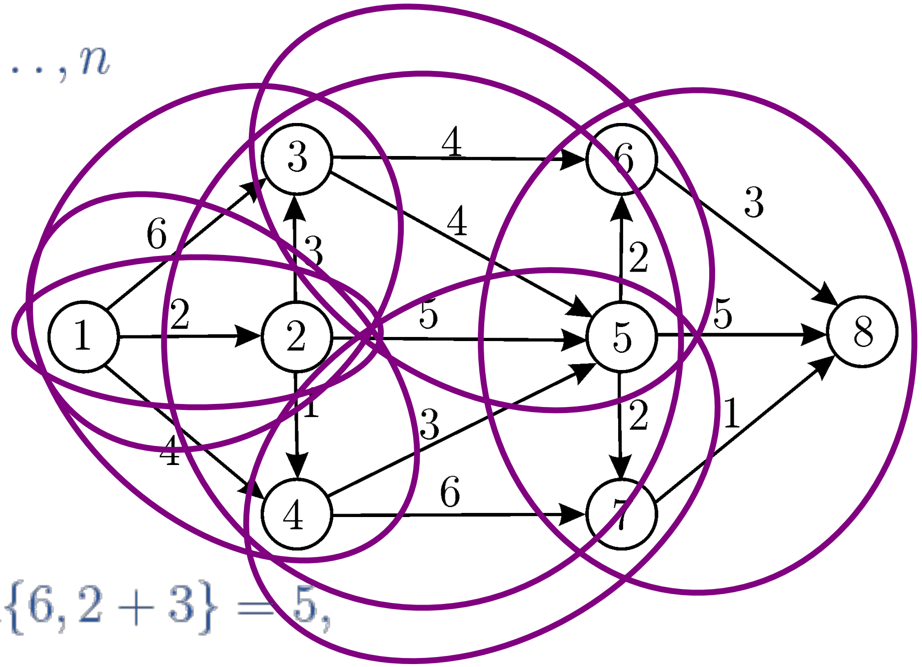


With the new numbering we can use Bellman's equations:

$$u_j = \min_{k<j,\ v_k \in \Gamma^{-1}(v_j)} \{u_k + \omega_{kj}\}, \quad j = 2, \ldots, n$$

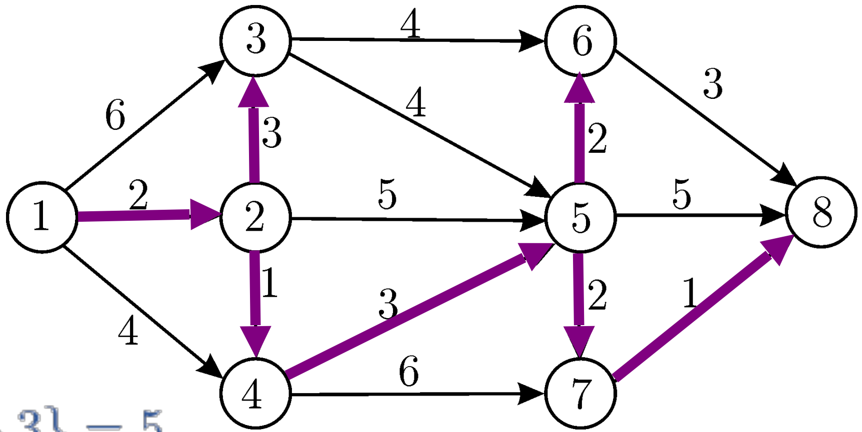
3. Acyclic Graphs. Critical Path Method.

$$u_j \;=\; \min_{k<j,\ v_k\in\Gamma^{-1}(v_j)} \{u_k+\omega_{kj}\}, \quad j=2,\ldots,n$$



EXAMPLE:

$$
\begin{aligned}
u_1 &= 0,\\
u_2 &= \min\{u_1+\omega_{12}\}=2,\\
u_3 &= \min\left\{u_1+\omega_{13},\underline{u_2+\omega_{23}}\right\}=\min\{6,2+3\}=5,\\
u_4 &= \min\left\{u_1+\omega_{14},\underline{u_2+\omega_{24}}\right\}=\min\{4,2+1\}=3,\\
u_5 &= \min\left\{u_2+\omega_{25},u_3+\omega_{35},\underline{u_4+\omega_{45}}\right\}=\min\{2+5,5+4,3+3\}=6,\\
u_6 &= \min\left\{u_3+\omega_{36},\underline{u_5+\omega_{56}}\right\}=\min\{5+4,6+2\}=8,\\
u_7 &= \min\left\{u_4+\omega_{47},\underline{u_5+\omega_{57}}\right\}=\min\{3+6,6+2\}=8,\\
u_8 &= \min\left\{u_5+\omega_{58},u_6+\omega_{68},\underline{u_7+\omega_{78}}\right\}=\min\{6+5,8+3,8+1\}=9.
\end{aligned}
$$



EXAMPLE:

BUILDING THE PATHS

$$
\begin{aligned}
u_1 &= 0,\\
u_2 &= \text{mín}\{u_1 + \omega_{12}\} = 2,\\
u_3 &= \text{mín}\left\{u_1 + \omega_{13}, \underline{u_2 + \omega_{23}}\right\} = \text{mín}\{6, 2+3\} = 5,\\
u_4 &= \text{mín}\left\{u_1 + \omega_{14}, \underline{u_2 + \omega_{24}}\right\} = \text{mín}\{4, 2+1\} = 3,\\
u_5 &= \text{mín}\left\{u_2 + \omega_{25}, u_3 + \omega_{35}, \underline{u_4 + \omega_{45}}\right\} = \text{mín}\{2+5, 5+4, 3+3\} = 6,\\
u_6 &= \text{mín}\left\{u_3 + \omega_{36}, \underline{u_5 + \omega_{56}}\right\} = \text{mín}\{5+4, 6+2\} = 8,\\
u_7 &= \text{mín}\left\{u_4 + \omega_{47}, \underline{u_5 + \omega_{57}}\right\} = \text{mín}\{3+6, 6+2\} = 8,\\
u_8 &= \text{mín}\left\{u_5 + \omega_{58}, u_6 + \omega_{68}, \underline{u_7 + \omega_{78}}\right\} = \text{mín}\left\{6+5, 8+3, 8+1\right\} = 9.
\end{aligned}
$$

3. Acyclic Graphs. Critical Path Method.

APPLICATION: PERT (Program (or Project) Evaluation and Review Technique)

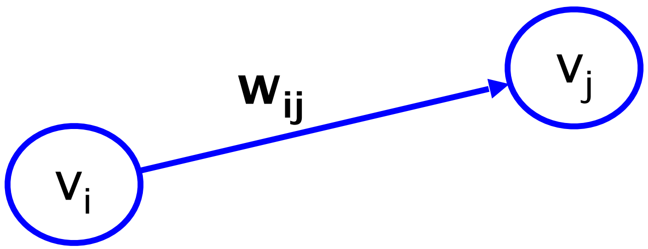
Program Evaluation and Review Technique (PERT) is used to **schedule** the tasks of a **large complicated project**.

PERT came into play during the 1950s in order to handle the complexities that arose in organizing the many individual activities required for the completion of projects on a very large scale. This technique was actually developed and first used by the U.S. Navy in order to coordinate the many projects that were necessary for the building of the Polaris submarine.

3. Acyclic Graphs. Critical Path Method.

APPLICATION: PERT (Program Evaluation and Review Technique)

- Represent the project using a directed graph.
- Each task of the project is represented by a vertex $\mathbf{v_j}$.
- If task $\mathbf{v_i}$ must be performed immediately before the start of task $\mathbf{v_j}$ we include an arc $(\mathbf{v_i},\mathbf{v_j})$.
- To this arc we will assign a weight $\mathbf{w_{ij}}$, representing the time between the start of the task $\mathbf{v_i}$ and the start of the task $\mathbf{v_j}$.
- The graph constructed in this way is acyclic since the existence of a circuit implies that the project is not feasible.

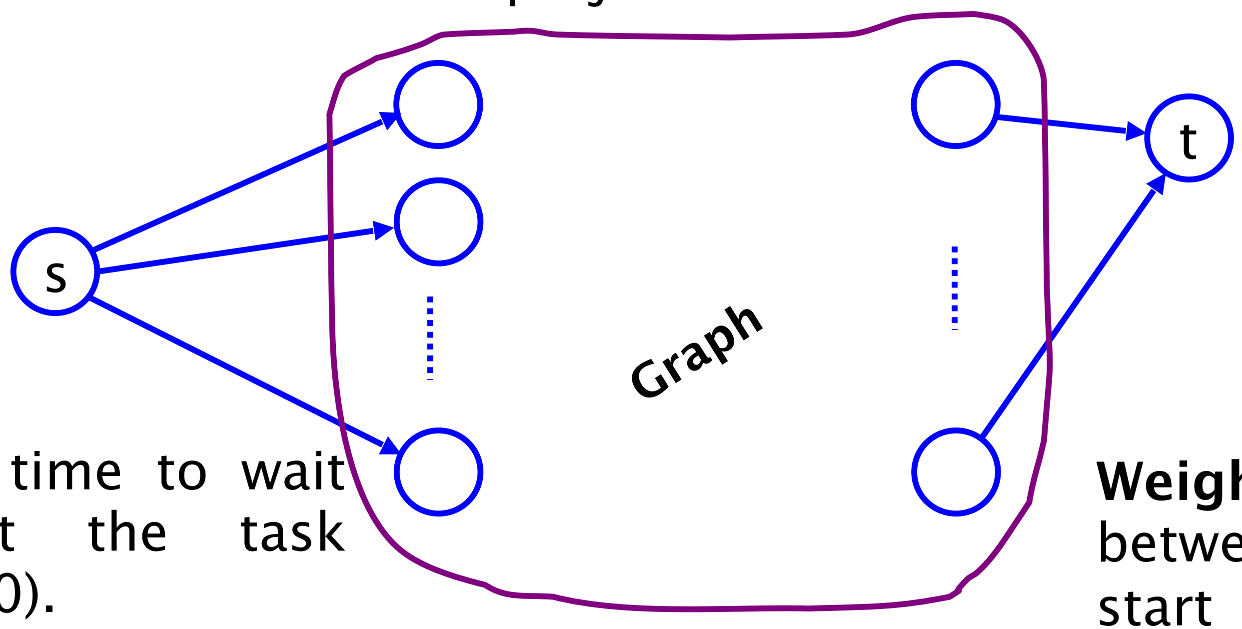


3. Acyclic Graphs. Critical Path Method.

APPLICATION: PERT (Program Evaluation and Review Technique)

We add a **fictitious** vertex s joining vertices with indegree zero. It indicates the **start** of the project.

We add a **fictitious** vertex t joining vertex with outdegree zero. It indicates the **end** of the project.



Weight: time to wait to start the task (usually 0).

Weight: time elapsed between the task start and the task end.

3. Acyclic Graphs. Critical Path Method.

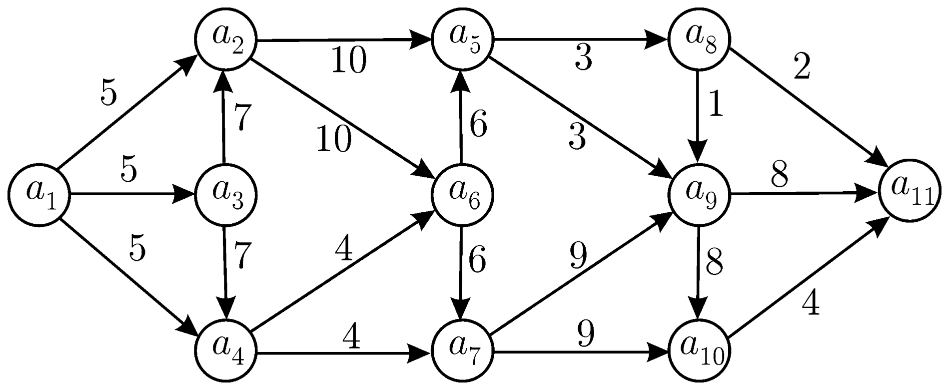
APPLICATION: PERT (Program Evaluation and Review Technique)

- Minimum time required to complete the entire project: weight of the **longest path** from s to t.
- This path is called the **critical path** since the included tasks determine the total time needed to complete the project and any delay in the execution of one of them involves a delay in project completion.
- By this reason these tasks are called **critical tasks**.
- How to calculate the minimum time and the critical path?

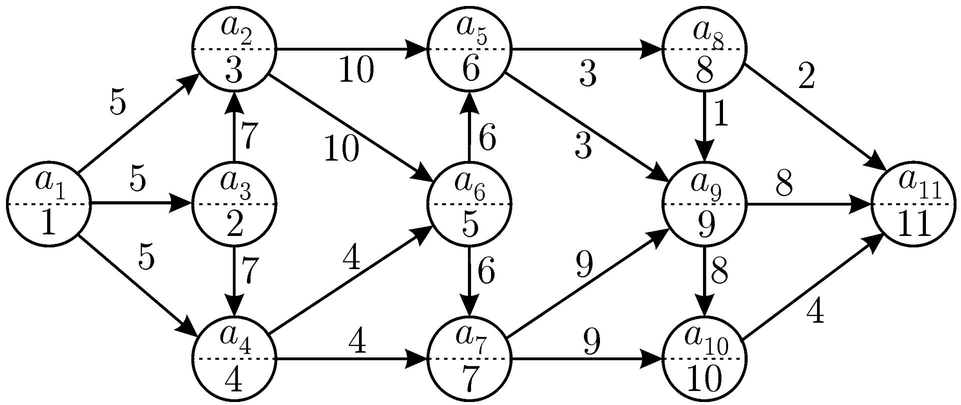
$$
\begin{aligned}
u_1 &= 0, \\
u_j &= \max_{k<j,\ v_k \in \Gamma^{-1}(v_j)} \{u_k + \omega_{kj}\}, \quad j = 2, \ldots, n,
\end{aligned}
$$

EXAMPLE 1: PERT

Calculate the minimum number of days needed to complete the next project.

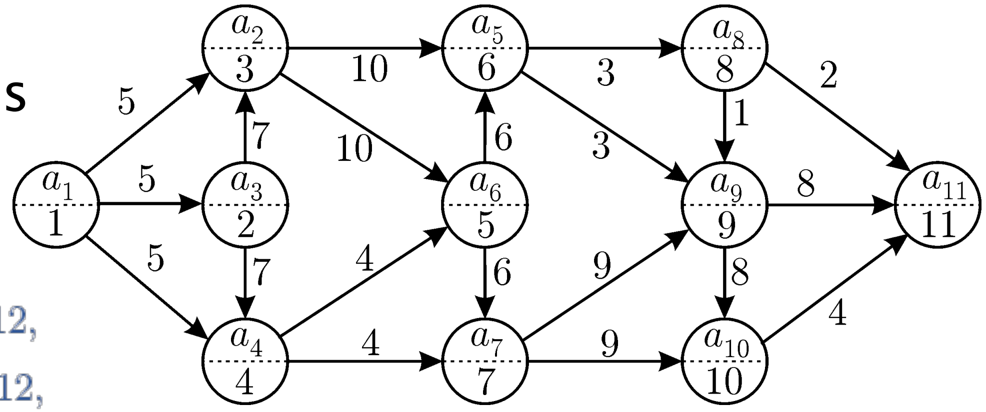


First: compute new numbering.



EXAMPLE 1: PERT

Second: Use Bellman's equations



$$
\begin{aligned}
u_1 &= 0,\\
u_2 &= \max\{u_1+\omega_{12}\} = 5,\\
u_3 &= \max\{u_1+\omega_{13}, \underline{u_2+\omega_{23}}\} = \max\{5, 5+7\} = 12,\\
u_4 &= \max\{u_1+\omega_{14}, \underline{u_2+\omega_{24}}\} = \max\{5, 5+7\} = 12,\\
u_5 &= \max\{\underline{u_3+\omega_{35}}, u_4+\omega_{45}\} = \max\{12+10, 12+4\} = 22,\\
u_6 &= \max\{u_3+\omega_{36}, \underline{u_5+\omega_{56}}\} = \max\{12+10, 22+6\} = 28,\\
u_7 &= \max\{u_4+\omega_{47}, \underline{u_5+\omega_{57}}\} = \max\{12+4, 22+6\} = 28,\\
u_8 &= \max\{u_6+\omega_{68}\} = 28+3 = 31,\\
u_9 &= \max\{u_6+\omega_{69}, \underline{u_7+\omega_{79}}, u_8+\omega_{89}\} = \max\{28+3, 28+9, 31+1\} = 37,\\
u_{10} &= \max\left\{u_7+\omega_{7,10}, \underline{u_9+\omega_{9,10}}\right\} = \max\{28+9, 37+8\} = 45,\\
u_{11} &= \max\left\{u_8+\omega_{8,11}, u_9+\omega_{9,11}, \underline{u_{10}+\omega_{10,11}}\right\} = \max\{31+2, 37+8, 45+4\} = 49.
\end{aligned}
$$

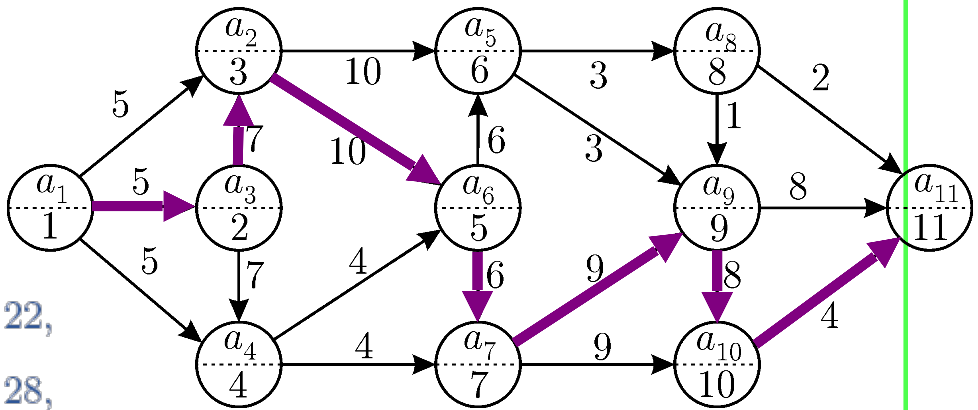
Minimum number of days needed to complete the project: 49

3. Acyclic Graphs. Critical Path Method.

EXAMPLE 1: PERT

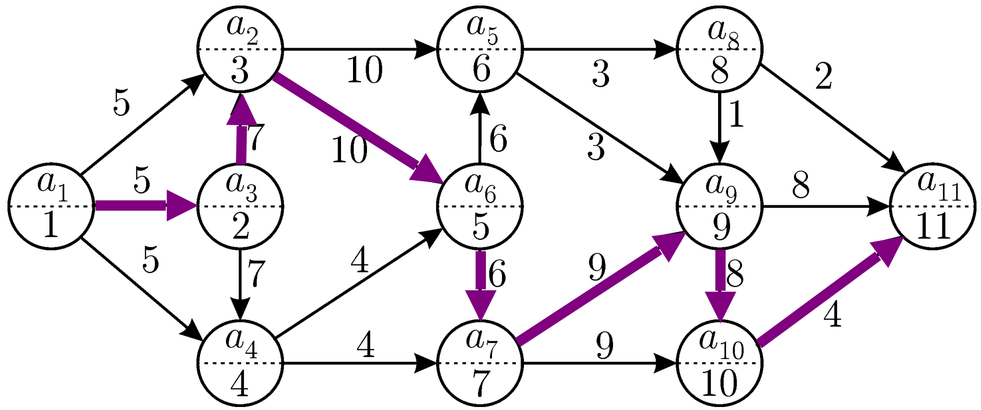
Third: Build the critical path

$$
\begin{aligned}
u_1 &= 0,\\
u_2 &= \text{máx}\,\{u_1+\omega_{12}\} = 5,\\
u_3 &= \text{máx}\,\{u_1+\omega_{13}, \underline{u_2+\omega_{23}}\} = \text{máx}\{5, 5+7\} = 12,\\
u_4 &= \text{máx}\,\{u_1+\omega_{14}, \underline{u_2+\omega_{24}}\} = \text{máx}\{5, 5+7\} = 12,\\
u_5 &= \text{máx}\,\{\underline{u_3+\omega_{35}}, u_4+\omega_{45}\} = \text{máx}\{12+10, 12+4\} = 22,\\
u_6 &= \text{máx}\,\{u_3+\omega_{36}, \underline{u_5+\omega_{56}}\} = \text{máx}\{12+10, 22+6\} = 28,\\
u_7 &= \text{máx}\,\{u_4+\omega_{47}, \underline{u_5+\omega_{57}}\} = \text{máx}\{12+4, 22+6\} = 28,\\
u_8 &= \text{máx}\,\{u_6+\omega_{68}\} = 28+3 = 31,\\
u_9 &= \text{máx}\,\{u_6+\omega_{69}, \underline{u_7+\omega_{79}}, u_8+\omega_{89}\} = \text{máx}\{28+3, 28+9, 31+1\} = 37,\\
u_{10} &= \text{máx}\left\{u_7+\omega_{7,10}, \underline{u_9+\omega_{9,10}}\right\} = \text{máx}\{28+9, 37+8\} = 45,\\
u_{11} &= \text{máx}\left\{u_8+\omega_{8,11}, u_9+\omega_{9,11}, \underline{u_{10}+\omega_{10,11}}\right\} = \text{máx}\{31+2, 37+8, 45+4\} = 49.
\end{aligned}
$$



Minimum number of days needed to complete the project: 49

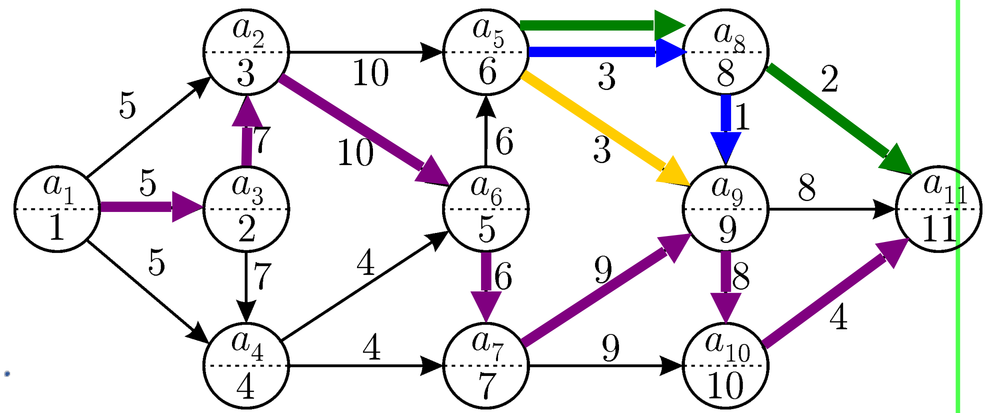
EXAMPLE 1: PERT



Question: Compute the maximum delay allowed for task a_5 (corresponding to renumbered vertex 6) without affecting the duration of the entire project.

Solution: consider the different paths that link the activity a_5 with the critical path.

EXAMPLE 1: PERT



- $P^{(1)}_{6,9}$: 6 9, $\omega(P^{(1)}_{6,9}) = 3.$
- $P^{(2)}_{6,9}$: 6 8 9, $\omega(P^{(2)}_{6,9}) = 4.$
- $P_{6,11}$: 6 8 11, $\omega(P_{6,11}) = 5.$

Suppose activity a_5 is delayed x days. The three previous paths must satisfy:

$$\left.\begin{array}{ll} P^{(1)}_{6,9}: & u_6 + \omega(P^{(1)}_{6,9}) + x \leq u_9 \\ P^{(2)}_{6,9}: & u_6 + \omega(P^{(2)}_{6,9}) + x \leq u_9 \\ P_{6,11}: & u_6 + \omega(P_{6,11}) + x \leq u_{11} \end{array}\right\} \Rightarrow \left.\begin{array}{l} 28+3+x \leq 37 \\ 28+4+x \leq 37 \\ 28+5+x \leq 49 \end{array}\right\} \Rightarrow \left.\begin{array}{l} x \leq 6 \\ x \leq 5 \\ x \leq 16 \end{array}\right\} \Rightarrow x \leq 5$$

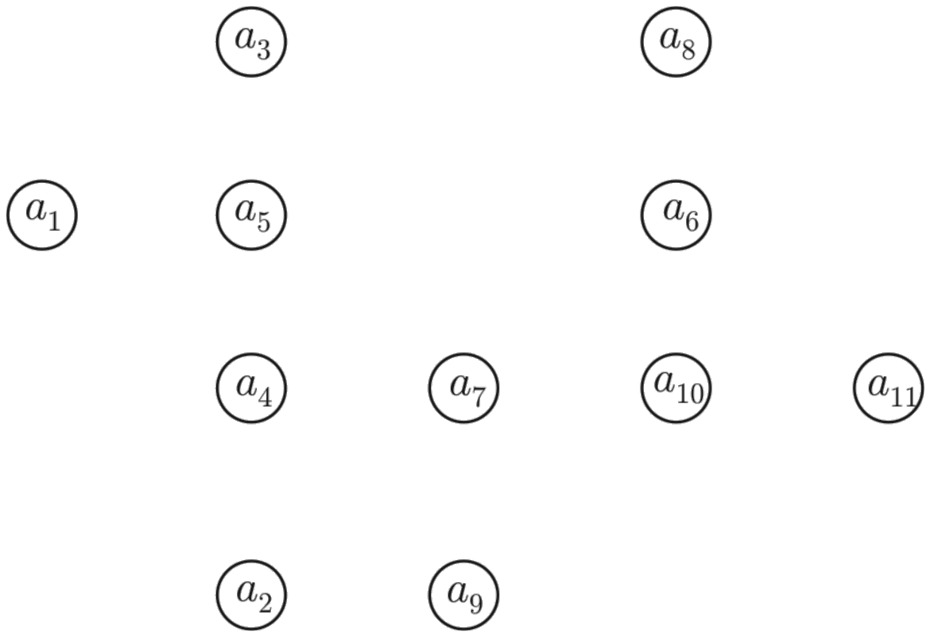
Maximum delay allowed for activity a_5: 5 days

EXAMPLE 2: PERT

How to represent the graph from a table of prerequisites?

Activity	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
Needed time	6	2	10	1	4	2	4	7	9	2	4
Prerequisites	–	–	a_1	a_1	a_1	a_5 a_{10}	a_2 a_4	a_3 a_6	a_2 a_4	a_7	a_8 a_{10}

First: Represent each activity using a vertex

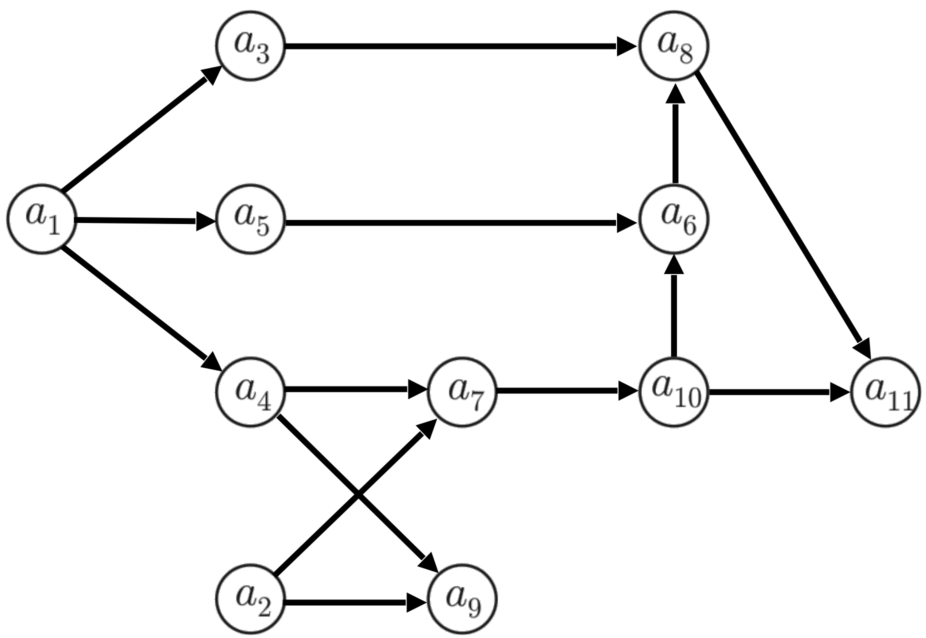


EXAMPLE 2: PERT

How to represent the graph from a table of prerequisites?

Activity	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
Needed time	6	2	10	1	4	2	4	7	9	2	4
Prerequisites	–	–	a_1	a_1	a_1	a_5 a_{10}	a_2 a_4	a_3 a_6	a_2 a_4	a_7	a_8 a_{10}

Second: Draw the arcs from the prerequisits.

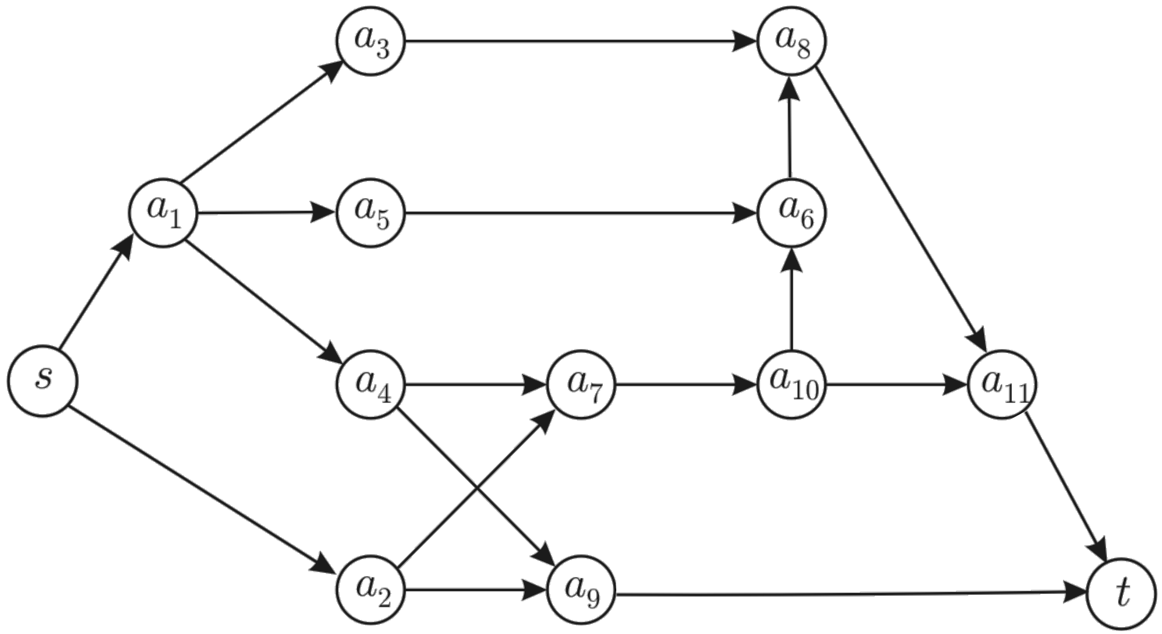


EXAMPLE 2: PERT

How to represent the graph from a table of prerequisites?

Activity	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
Needed time	6	2	10	1	4	2	4	7	9	2	4
Prerequisites	–	–	a_1	a_1	a_1	a_5 a_{10}	a_2 a_4	a_3 a_6	a_2 a_4	a_7	a_8 a_{10}

Third: Add fictitious vertices



3. Acyclic Graphs. Critical Path Method.

EXAMPLE 2: PERT

How to represent the graph from a table of prerequisites?

Activity	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
Needed time	6	2	10	1	4	2	4	7	9	2	4
Prerequisites	–	–	a_1	a_1	a_1	a_5 a_{10}	a_2 a_4	a_3 a_6	a_2 a_4	a_7	a_8 a_{10}

Forth: Add weights