UNIVERSIDAD NACIONAL DE COLOMBIA

Algoritmia Avanzada

Sesión 7

All-Pairs Shortest Path Algorithms

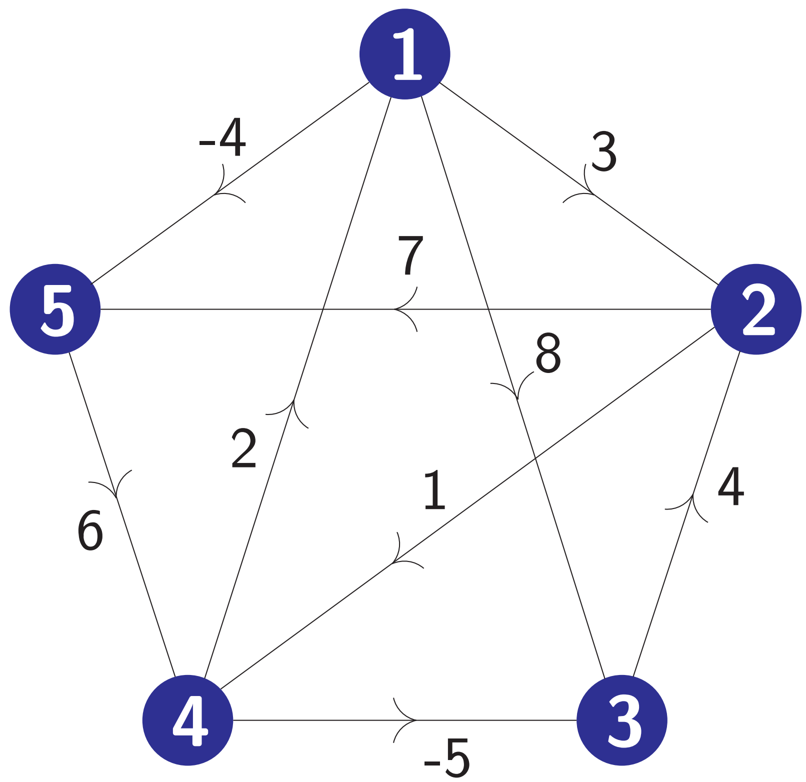
Yoan Pinzón, PhD
Universidad Nacional de Colombia
http://disi.unal.edu.co/~ypinzon/2019762/

Session 7

- **Shortest Paths**
  - ▷ APSP (All-Pairs Shortest Paths)
    - ◇ The Floyd-Warshall Algorithm
    - ◇ The Johnson's Algorithm

All-Pair Shortest Paths (APSP)

Compute the shortest paths for all pairs of vertices $u, v \in V$.



- $G = (V, E)$: directed graph; $n = |V|, m = |E|$
- $w(u, v)$: the weight of edge (u, v)

The Floyd-Warshall Algorithm

The Floyd-Warshall Algorithm solves the APSP problem and can also be use to report negative cycles.

- **Pseudo-code:**

```
FLOYD-WARSHALL(W)
    n ← rows[W]
    D^(0) ← W
    for k ← 1 to n do
        for i ← 1 to n do
            for j ← 1 to n do
                d_{i,j}^(k) ← min(d_{i,j}^(k-1), d_{i,k}^(k-1) + d_{k,j}^(k-1))
    return D^(n)
```

- **Time Complexity:** $\Theta(n^3)$

The Floyd-Warshall Algorithm

$d_{i,j}^{(k)}$ is the minimum weight of a path from vertex i to vertex j with all intermediate vertices in the set $\{1, 2, \ldots, k\}$.

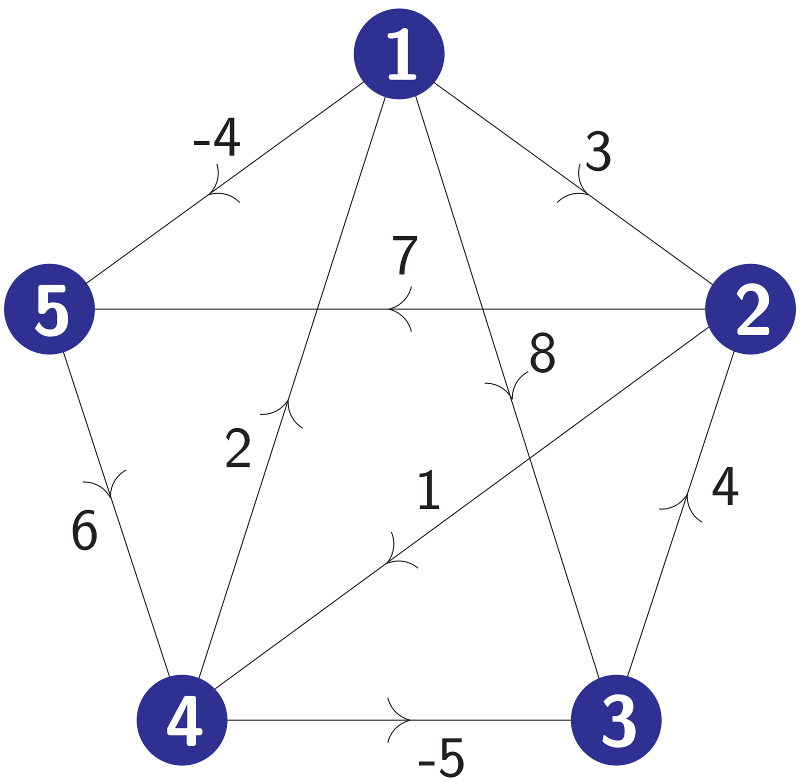
$$d_{i,j}^{(0)} = \begin{cases} 0 & \text{, if } \; i = j \\ w(i,j) & \text{, if } \; (i,j) \in E \\ \infty & \text{, otherwise} \end{cases}$$

$$d_{i,j}^{(k)} = min(d_{i,j}^{(k-1)}, d_{i,k}^{(k-1)} + d_{k,j}^{(k-1)}) \text{ for } k \geq 1$$

There is a negative cycle iff $d_{i,i}^{(n)} < 0$ for some i

The Floyd-Warshall Algorithm
Example

Compute the APSP for the following digraph $G = (V, E)$.



$$W = D^{(0)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

The Floyd-Warshall Algorithm
Solution

$$D^{(0)} = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

$$d_{1,2}^{(1)} = min(d_{1,2}^{(0)}, d_{1,1}^{(0)} + d_{1,2}^{(0)}) = min(3, 0+3) = min(3,3) = 3$$
$$d_{1,3}^{(1)} = min(d_{1,3}^{(0)}, d_{1,1}^{(0)} + d_{1,3}^{(0)}) = min(8, 0+8) = min(8,8) = 8$$
$$d_{1,4}^{(1)} = min(d_{1,4}^{(0)}, d_{1,1}^{(0)} + d_{1,4}^{(0)}) = min(\infty, 0+\infty) = min(\infty,\infty) = \infty$$
$$d_{1,5}^{(1)} = min(d_{1,5}^{(0)}, d_{1,1}^{(0)} + d_{1,5}^{(0)}) = min(-4, 0+-4) = min(-4,-4) = -4$$
$$d_{2,1}^{(1)} = min(d_{2,1}^{(0)}, d_{2,1}^{(0)} + d_{1,1}^{(0)}) = min(\infty, \infty+0) = min(\infty,\infty) = \infty$$
$$d_{2,3}^{(1)} = min(d_{2,3}^{(0)}, d_{2,1}^{(0)} + d_{1,3}^{(0)}) = min(\infty, \infty+8) = min(\infty,\infty) = \infty$$
$$d_{2,4}^{(1)} = min(d_{2,4}^{(0)}, d_{2,1}^{(0)} + d_{1,4}^{(0)}) = min(1, \infty+\infty) = min(1,\infty) = 1$$
$$d_{2,5}^{(1)} = min(d_{2,5}^{(0)}, d_{2,1}^{(0)} + d_{1,5}^{(0)}) = min(7, \infty+-4) = min(7,\infty) = 7$$
$$d_{3,1}^{(1)} = min(d_{3,1}^{(0)}, d_{3,1}^{(0)} + d_{1,1}^{(0)}) = min(\infty, \infty+0) = min(\infty,\infty) = \infty$$
$$d_{3,2}^{(1)} = min(d_{3,2}^{(0)}, d_{3,1}^{(0)} + d_{1,2}^{(0)}) = min(4, \infty+3) = min(4,\infty) = 4$$

The Floyd-Warshall Algorithm
Solution

$$D^{(0)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

$d_{3,4}^{(1)} = min(d_{3,4}^{(0)}, d_{3,1}^{(0)} + d_{1,4}^{(0)}) = min(\infty, \infty + \infty) = min(\infty, \infty) = \infty$

$d_{3,5}^{(1)} = min(d_{3,5}^{(0)}, d_{3,1}^{(0)} + d_{1,5}^{(0)}) = min(\infty, \infty + -4) = min(\infty, \infty) = \infty$

$d_{4,1}^{(1)} = min(d_{4,1}^{(0)}, d_{4,1}^{(0)} + d_{1,1}^{(0)}) = min(2, 2 + 0) = min(2, 2) = 2$

$d_{4,2}^{(1)} = min(d_{4,2}^{(0)}, d_{4,1}^{(0)} + d_{1,2}^{(0)}) = min(\infty, 2 + 3) = min(\infty, 5) = 5$ ★

$d_{4,3}^{(1)} = min(d_{4,3}^{(0)}, d_{4,1}^{(0)} + d_{1,3}^{(0)}) = min(-5, 2 + 8) = min(-5, 10) = -5$

$d_{4,5}^{(1)} = min(d_{4,5}^{(0)}, d_{4,1}^{(0)} + d_{1,5}^{(0)}) = min(\infty, 2 + -4) = min(\infty, -2) = -2$ ★

$d_{5,1}^{(1)} = min(d_{5,1}^{(0)}, d_{5,1}^{(0)} + d_{1,1}^{(0)}) = min(\infty, \infty + 0) = min(\infty, \infty) = \infty$

$d_{5,2}^{(1)} = min(d_{5,2}^{(0)}, d_{5,1}^{(0)} + d_{1,2}^{(0)}) = min(\infty, \infty + 3) = min(\infty, \infty) = \infty$

$d_{5,3}^{(1)} = min(d_{5,3}^{(0)}, d_{5,1}^{(0)} + d_{1,3}^{(0)}) = min(\infty, \infty + 8) = min(\infty, \infty) = \infty$

$d_{5,4}^{(1)} = min(d_{5,4}^{(0)}, d_{5,1}^{(0)} + d_{1,4}^{(0)}) = min(6, \infty + \infty) = min(6, \infty) = 6$

The Floyd-Warshall Algorithm
Solution

$$D^{(0)} = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad D^{(1)} = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \mathbf{5} & -5 & 0 & \mathbf{-2} \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad D^{(3)} = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix} \qquad D^{(5)} = \begin{bmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix}$$

The Floyd-Warshall Algorithm

Solution

$\pi_{i,j}^{(k)}$ is the predecessor of vertex j on a shortest path from vertex i with all intermediate vertices in the set $\{1, 2, \ldots, k\}$.

$$\pi_{i,j}^{(0)} = \begin{cases} \emptyset & \text{, if } \; i = j \text{ or } w_{i,j} = \infty \\ i & \text{, if } \; i \neq j \text{ and } w_{i,j} < \infty \end{cases}$$

$$\pi_{i,j}^{(k)} = \begin{cases} \pi_{i,j}^{(k-1)} & \text{, if } \; d_{i,j}^{(k-1)} = d_{i,j}^{(k)} \\ \pi_{k,j}^{(k-1)} & \text{, if } \; d_{i,j}^{(k-1)} \neq d_{i,j}^{(k)} \end{cases}$$

The Floyd-Warshall Algorithm
Solution

$$D^{(0)} = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \Pi^{(0)} = \begin{bmatrix} \varnothing & 1 & 1 & \varnothing & 1 \\ \varnothing & \varnothing & \varnothing & 2 & 2 \\ \varnothing & 3 & \varnothing & \varnothing & \varnothing \\ 4 & \varnothing & 4 & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & 5 & \varnothing \end{bmatrix}$$

$$D^{(1)} = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \mathbf{5} & -5 & 0 & \mathbf{-2} \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \Pi^{(1)} = \begin{bmatrix} \varnothing & 1 & 1 & \varnothing & 1 \\ \varnothing & \varnothing & \varnothing & 2 & 2 \\ \varnothing & 3 & \varnothing & \varnothing & \varnothing \\ 4 & \mathbf{1} & 4 & \varnothing & \mathbf{1} \\ \varnothing & \varnothing & \varnothing & 5 & \varnothing \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & 3 & 8 & \mathbf{4} & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \mathbf{5} & \mathbf{11} \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \Pi^{(2)} = \begin{bmatrix} \varnothing & 1 & 1 & \mathbf{2} & 1 \\ \varnothing & \varnothing & \varnothing & 2 & 2 \\ \varnothing & 3 & \varnothing & \mathbf{2} & \mathbf{2} \\ 4 & 1 & 4 & \varnothing & 1 \\ \varnothing & \varnothing & \varnothing & 5 & \varnothing \end{bmatrix}$$

The Floyd-Warshall Algorithm
Solution

$$D^{(3)} = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & \mathbf{-1} & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \Pi^{(3)} = \begin{bmatrix} \varnothing & 1 & 1 & 2 & 1 \\ \varnothing & \varnothing & \varnothing & 2 & 2 \\ \varnothing & 3 & \varnothing & 2 & 2 \\ 4 & \mathbf{3} & 4 & \varnothing & 1 \\ \varnothing & \varnothing & \varnothing & 5 & \varnothing \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 3 & \mathbf{-1} & 4 & -4 \\ \mathbf{3} & 0 & \mathbf{-4} & 1 & \mathbf{-1} \\ \mathbf{7} & 4 & 0 & 5 & \mathbf{3} \\ 2 & -1 & -5 & 0 & -2 \\ \mathbf{8} & \mathbf{5} & \mathbf{1} & 6 & 0 \end{bmatrix} \qquad \Pi^{(4)} = \begin{bmatrix} \varnothing & 1 & \mathbf{4} & 2 & 1 \\ \mathbf{4} & \varnothing & \mathbf{4} & 2 & \mathbf{1} \\ \mathbf{4} & 3 & \varnothing & 2 & \mathbf{1} \\ 4 & 3 & 4 & \varnothing & 1 \\ \mathbf{4} & \mathbf{3} & \mathbf{4} & 5 & \varnothing \end{bmatrix}$$

$$D^{(5)} = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{-3} & \mathbf{2} & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix} \qquad \Pi^{(5)} = \begin{bmatrix} \varnothing & \mathbf{3} & \mathbf{4} & \mathbf{5} & 1 \\ 4 & \varnothing & 4 & 2 & 1 \\ 4 & 3 & \varnothing & 2 & 1 \\ 4 & 3 & 4 & \varnothing & 1 \\ 4 & 3 & 4 & 5 & \varnothing \end{bmatrix}$$

The Johnson's Algorithm

The Johnson's Algorithm solves the APSP problem by reweighing.

▸ **Pseudo-code:**

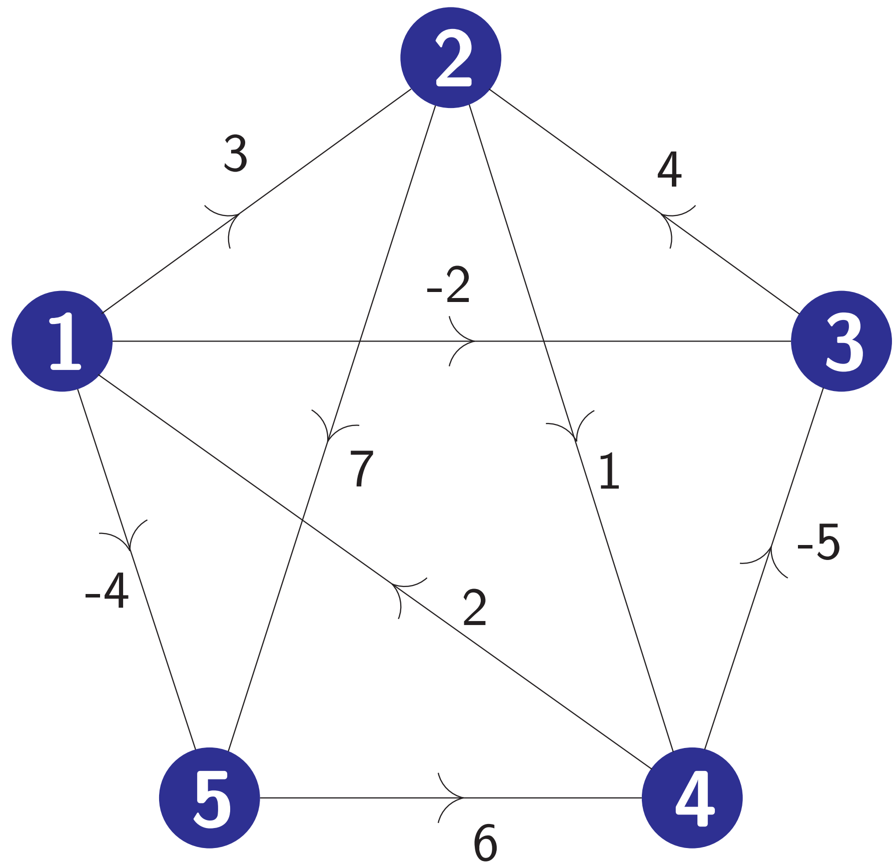
$\textsc{Johnson}(G, w)$

Compute $G' \leftarrow (V', E') : V' \leftarrow V \cup \{s\}, E' \leftarrow E \cup \{(s,v) | v \in V\}$
$w(s,v) \leftarrow 0 \;\; \forall\, v \in V$ ▷ Assign weight to new edges
if $\textsc{Bellman-Ford}(G', w, s) =$ FALSE **then**
  terminate ▷ There is a negative cycle in G
else
  for each $v \in V$ **do** $h[v] \leftarrow \delta(s,v)$
  for each $(u,v) \in E$ **do** $\widehat{w}(u,v) \leftarrow w(u,v) + h(u) - h(v)$
  for each $u \in V$ **do**
    run $\textsc{Dijkstra}(G, \widehat{w}, u)$ to compute $\widehat{\delta}(u,v) \;\forall\, v \in V$
    for each $v \in V$ **do** $d_{uv} \leftarrow \widehat{\delta}(u,v) - h(u) + h(v)$
return $D = d_{uv}$

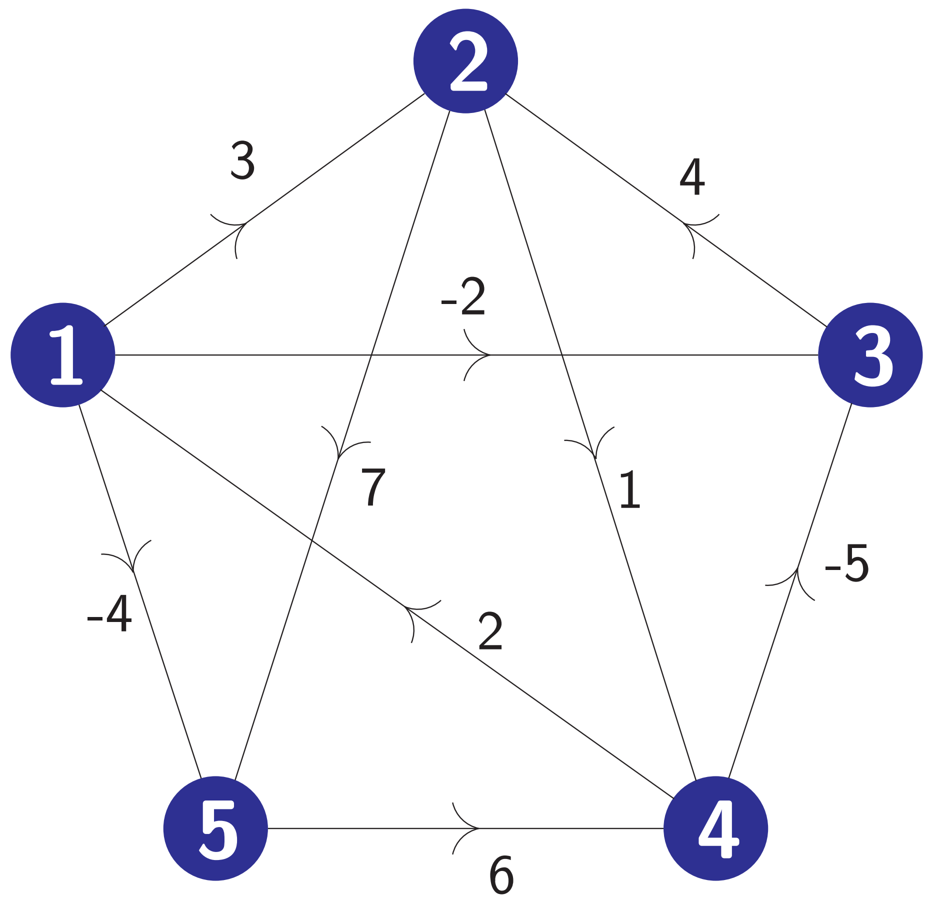
▸ **Time Complexity:** $\mathcal{O}(nm \log n)$

The Johnson's Algorithm
Example

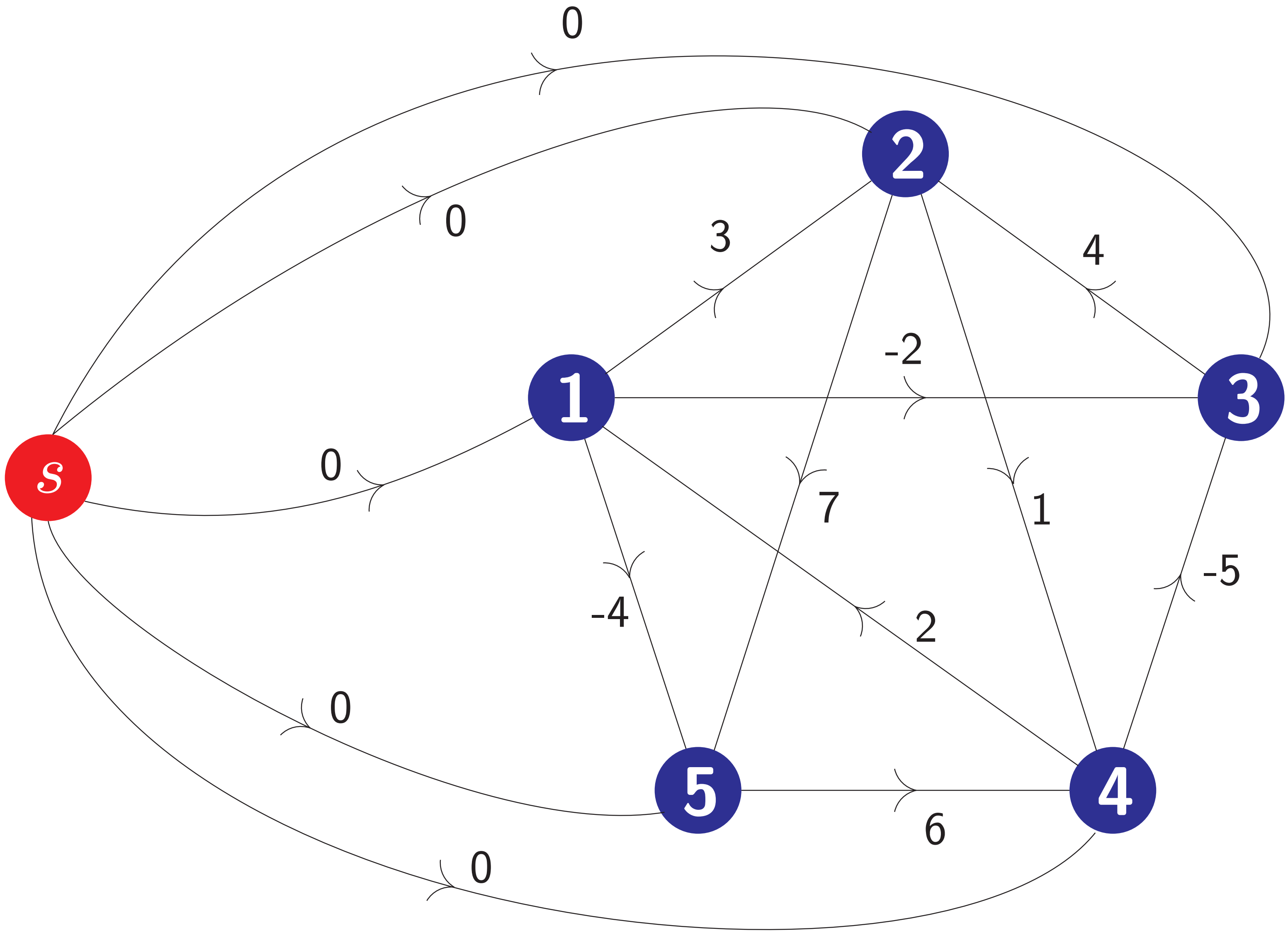
Compute the APSP for the following digraph $G = (V, E)$.



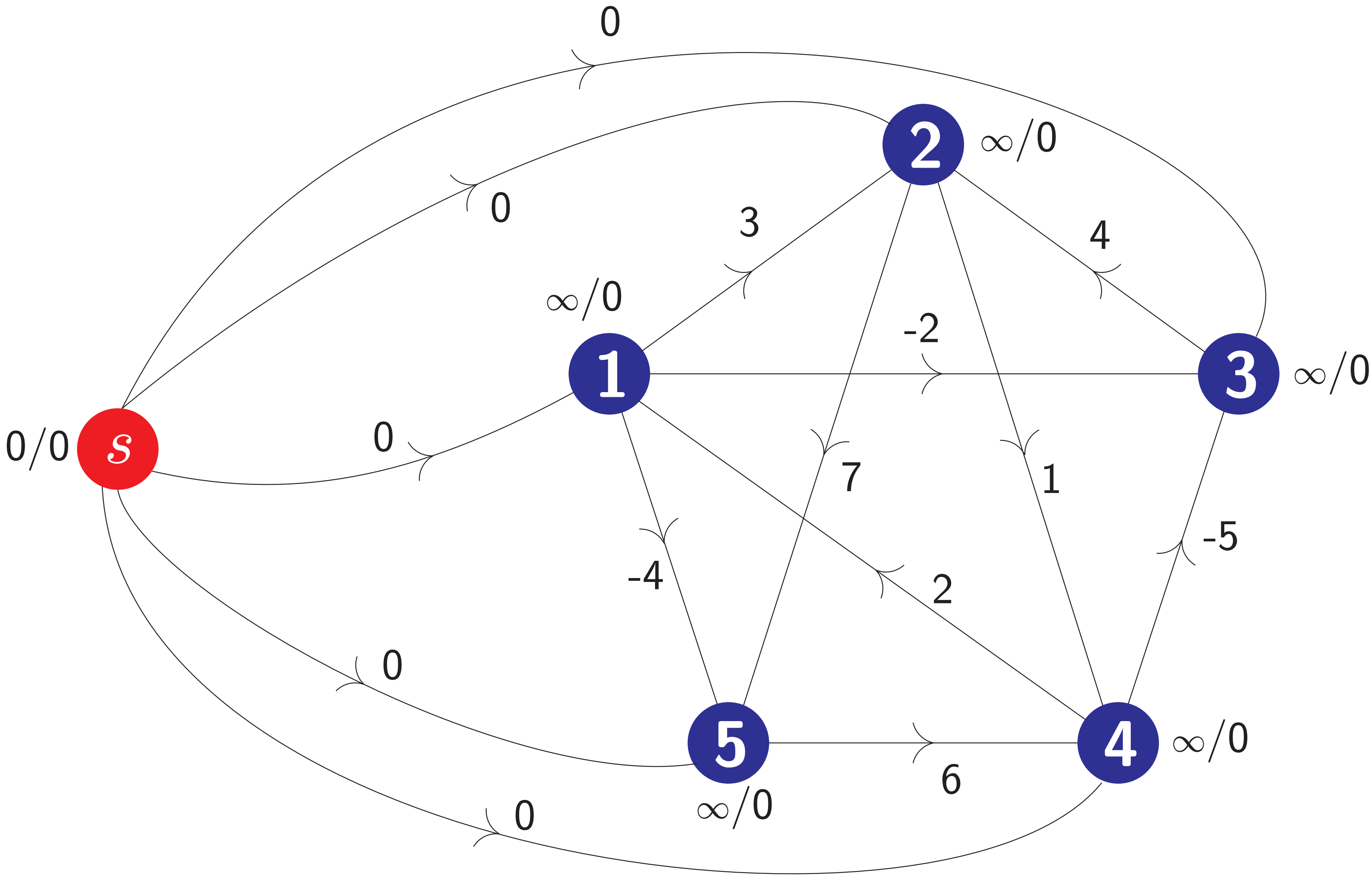
The Johnson's Algorithm
Solution



The Johnson's Algorithm
Solution

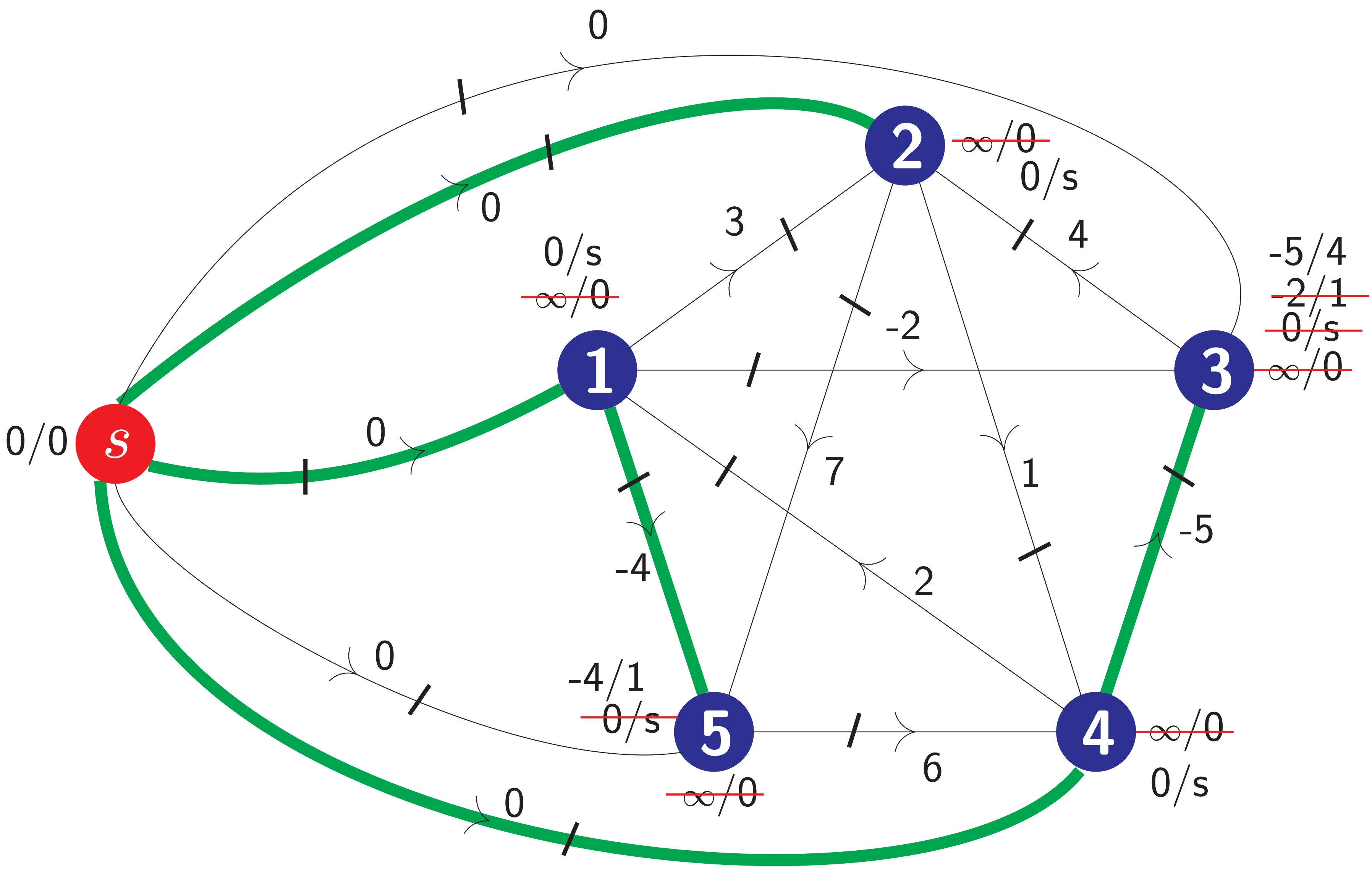


The Johnson's Algorithm
Solution

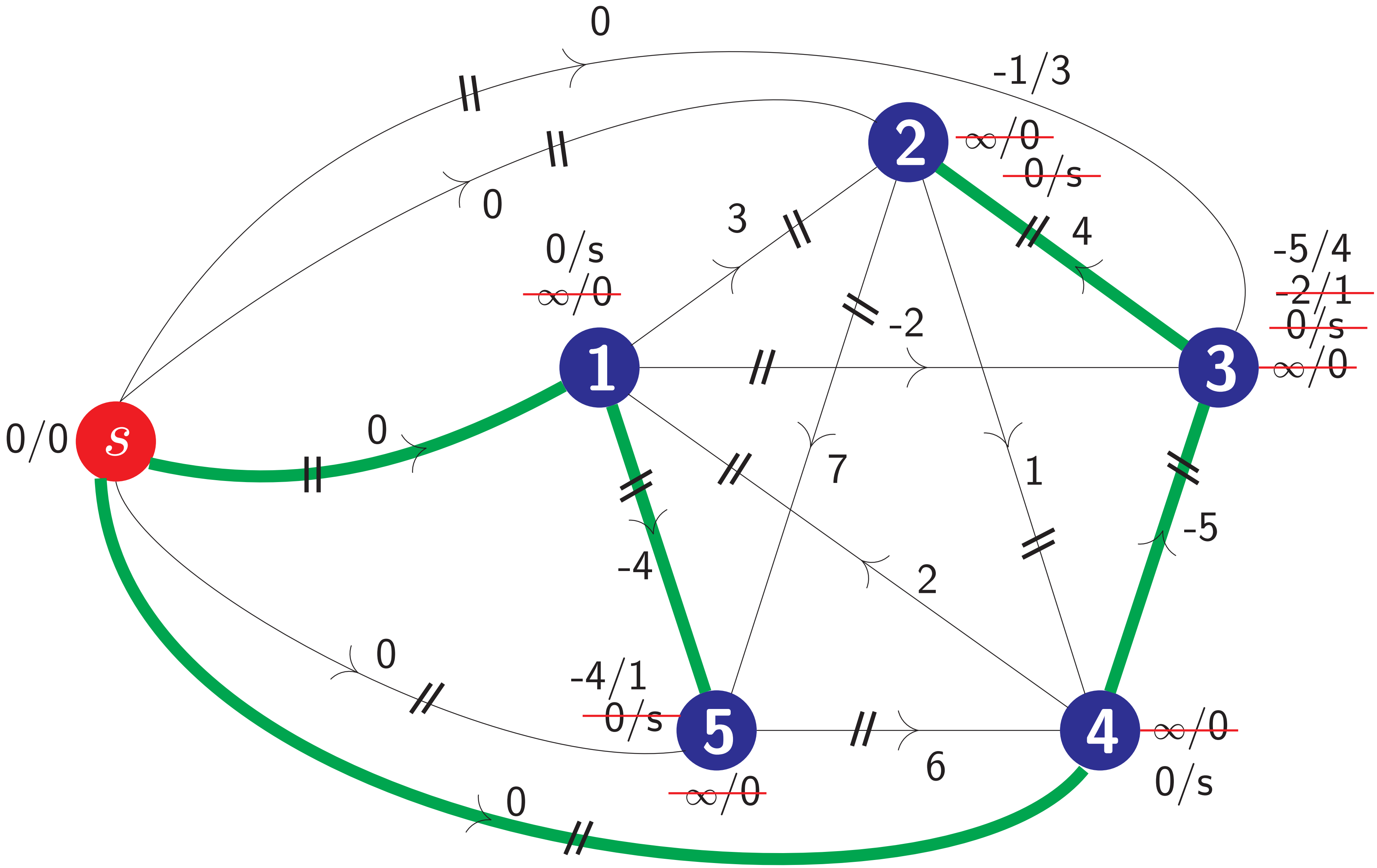


The Johnson's Algorithm

Solution

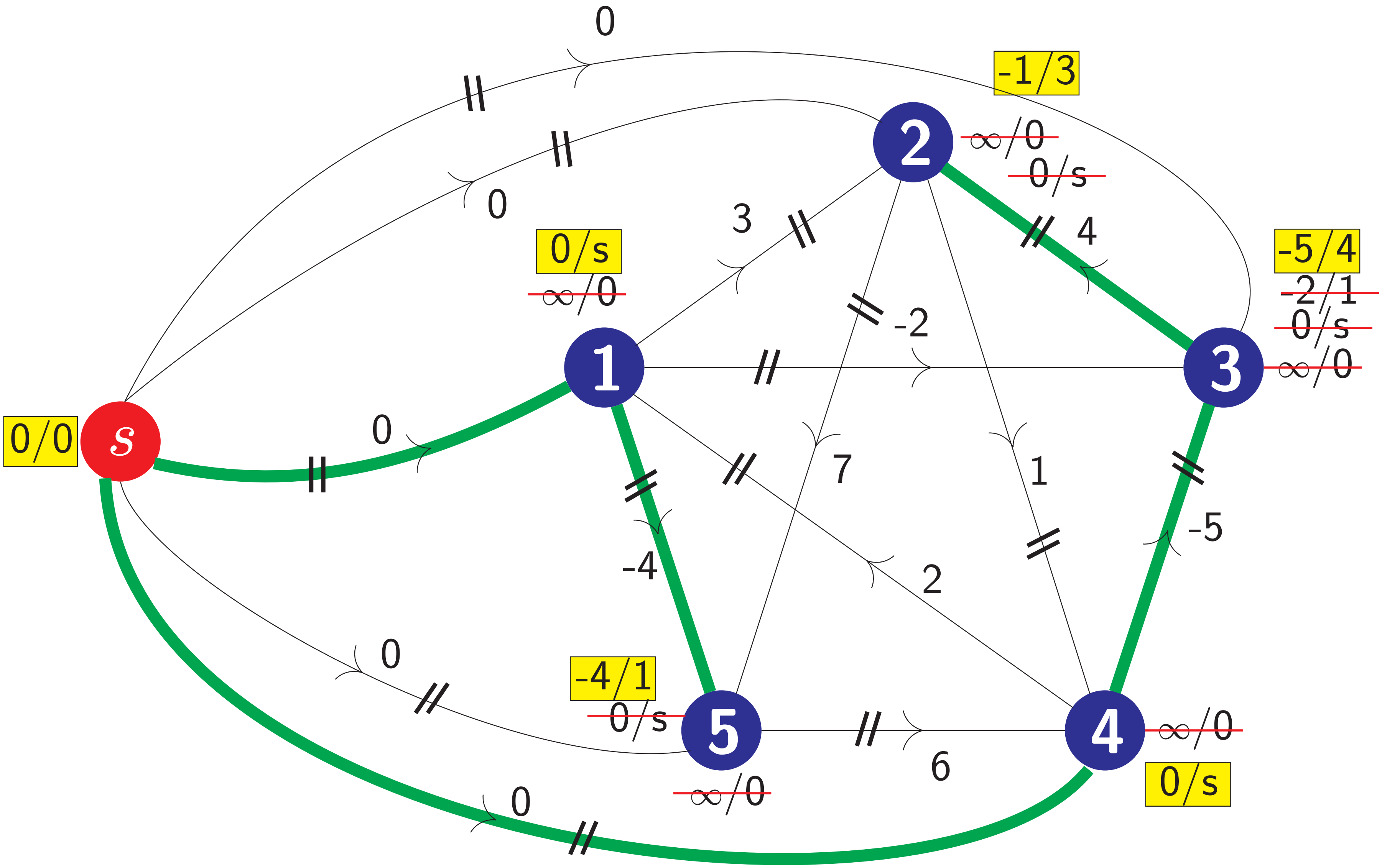


The Johnson's Algorithm
Solution



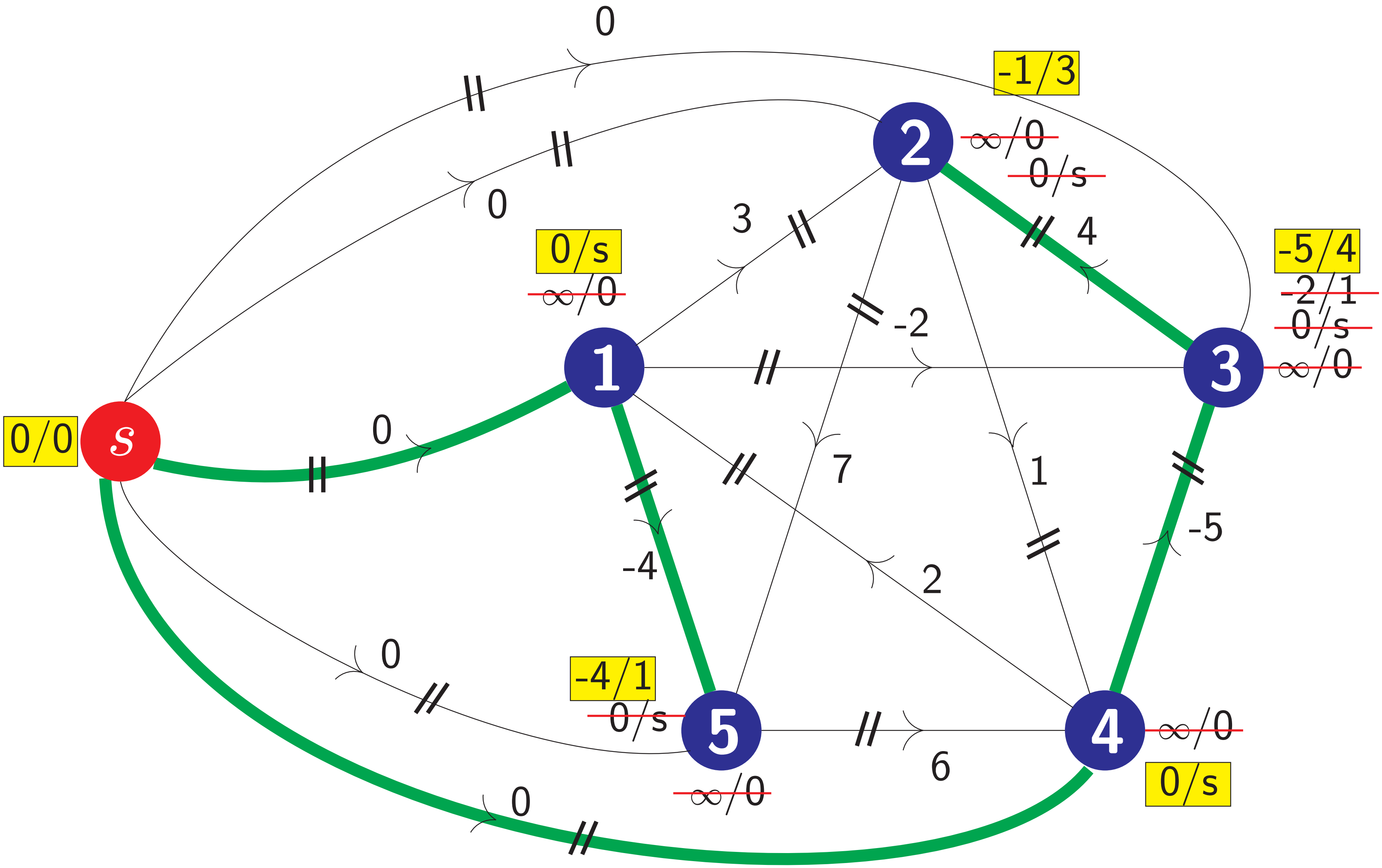
The Johnson's Algorithm

Solution



The Johnson's Algorithm

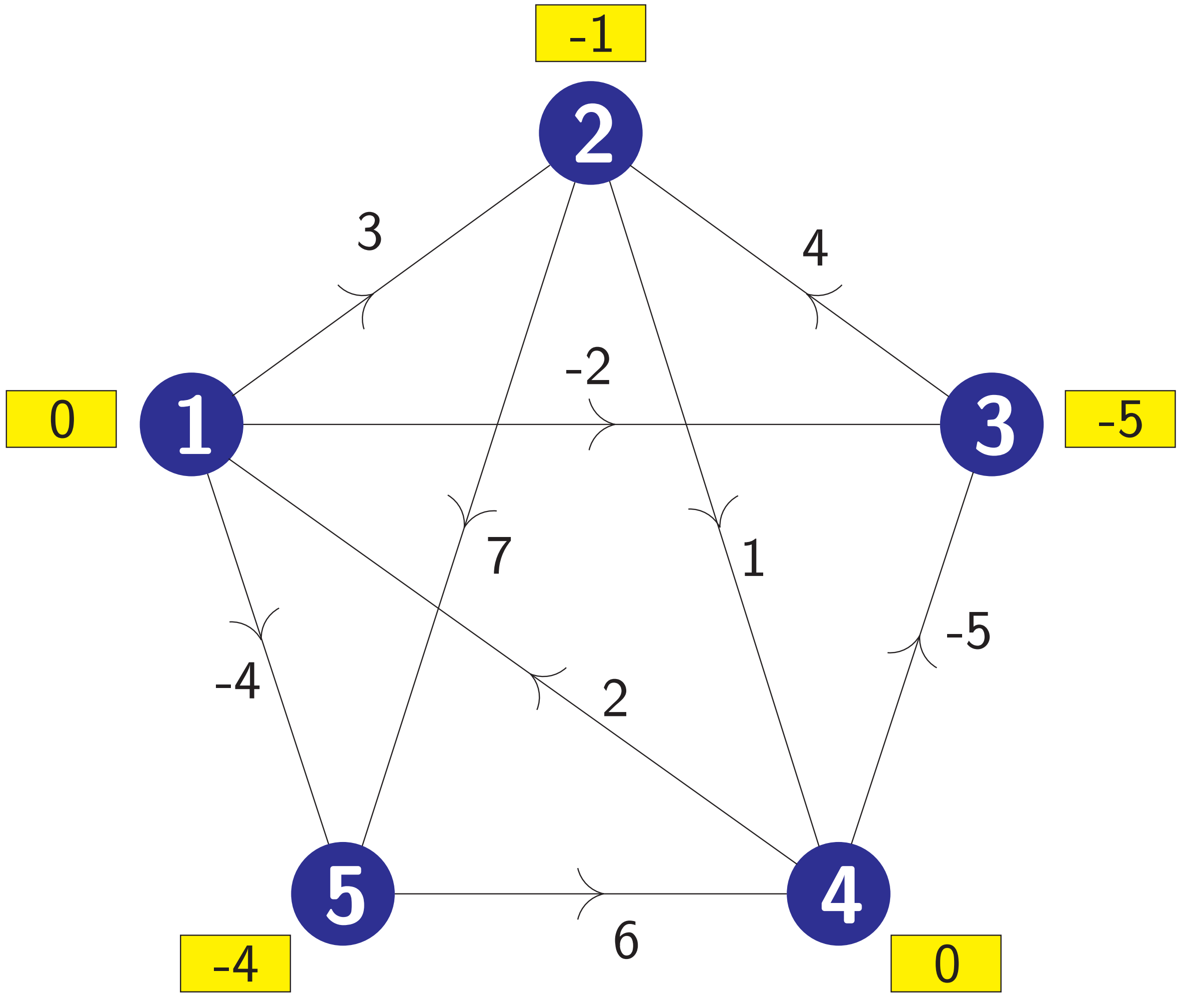
Solution



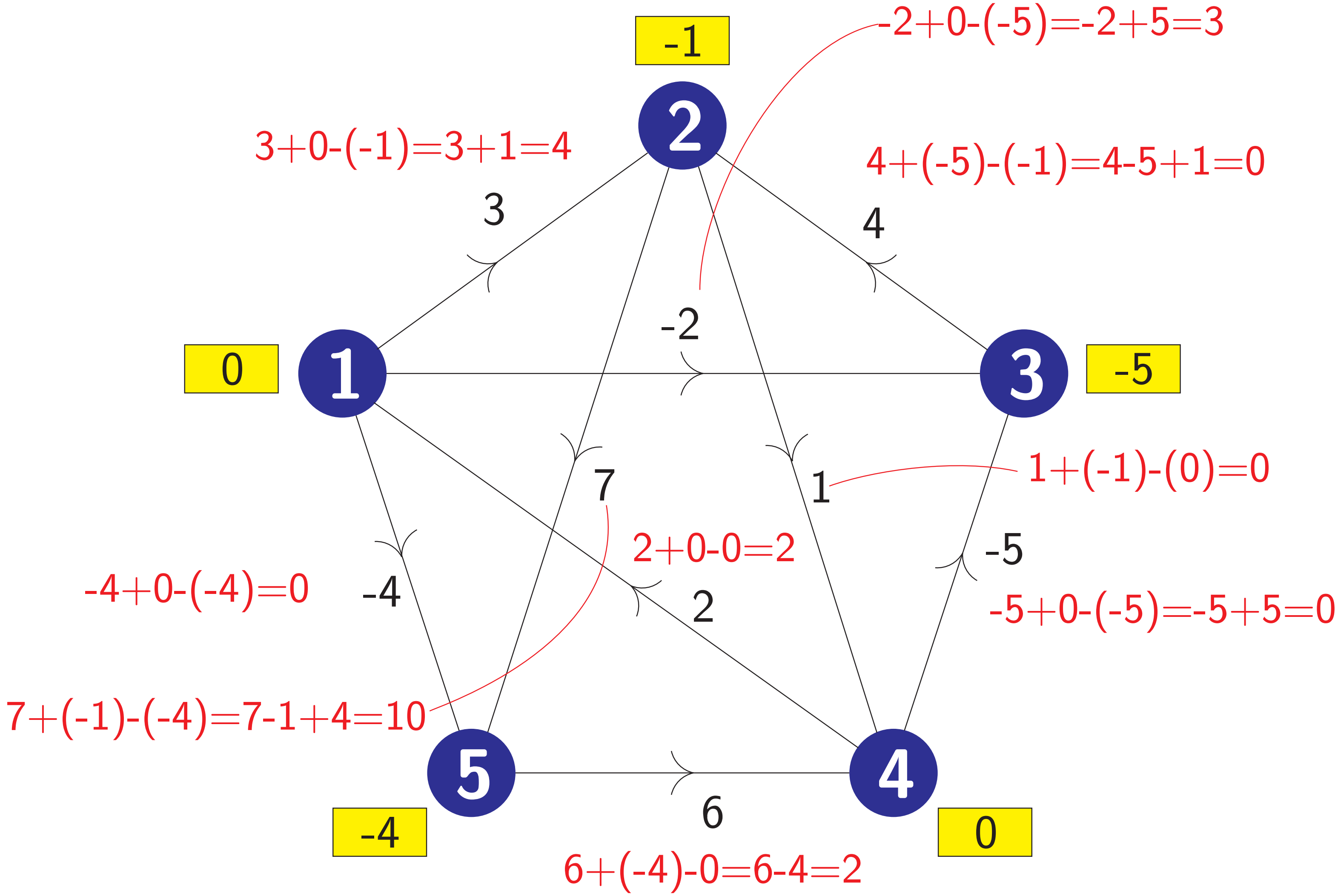
	s	1	2	3	4	5
$h=$	0	0	-1	-5	0	-4

$\delta(s,u)$

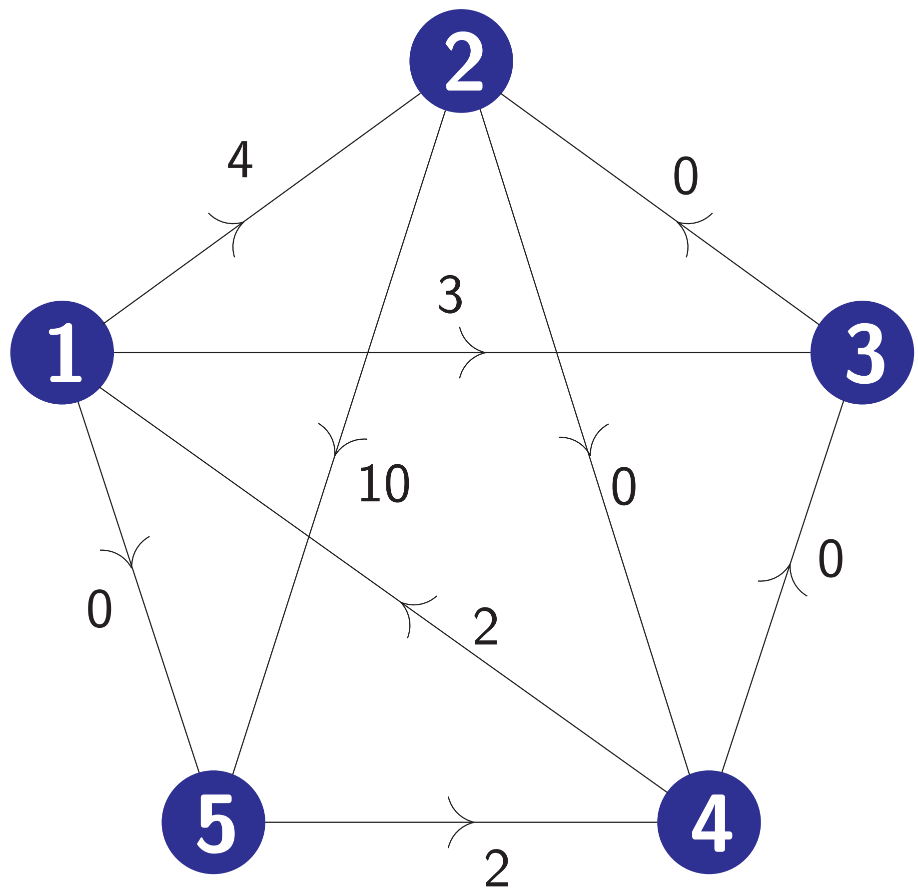
The Johnson's Algorithm
Solution



The Johnson's Algorithm
Solution

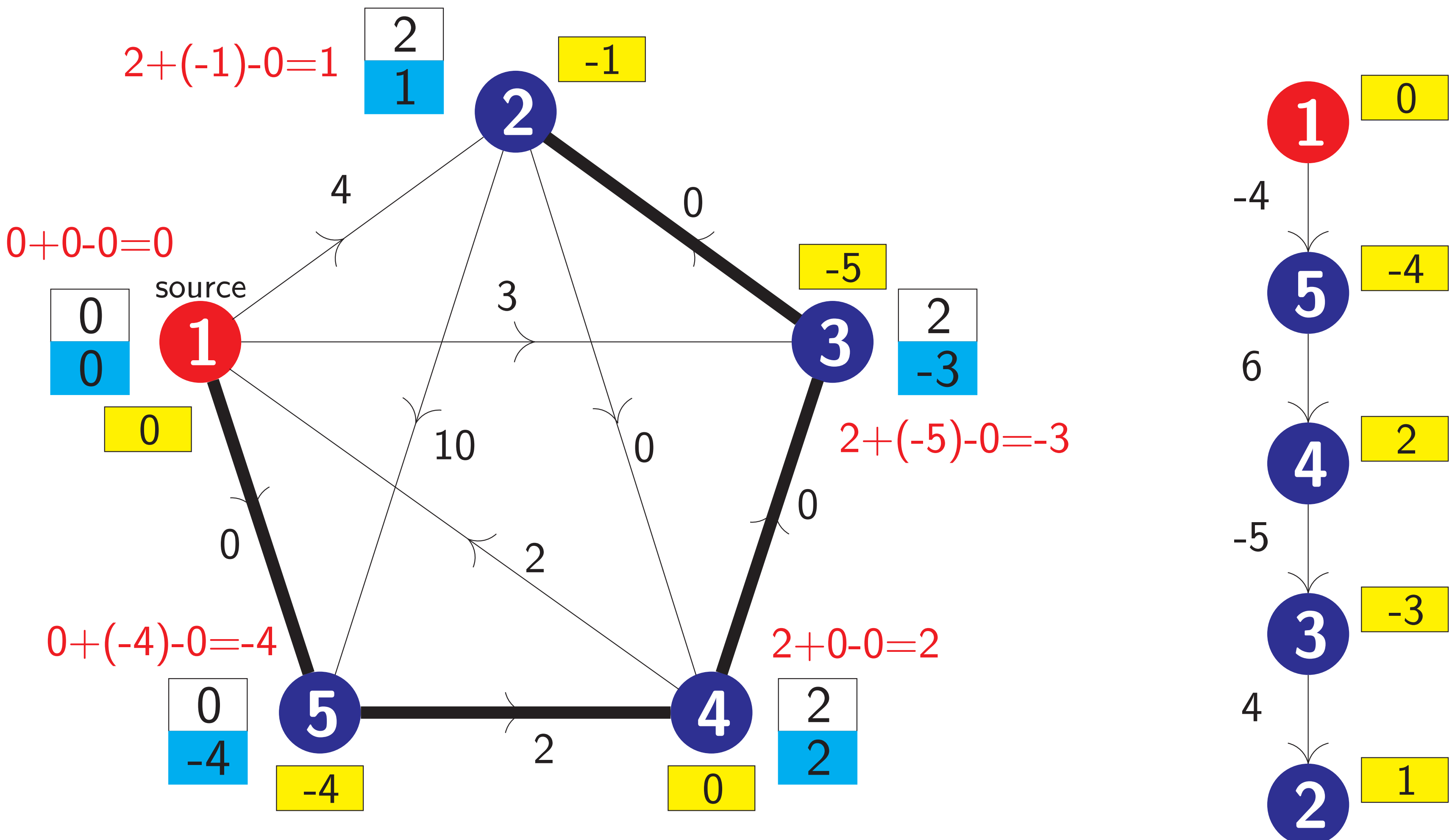


The Johnson's Algorithm
Solution



The Johnson's Algorithm

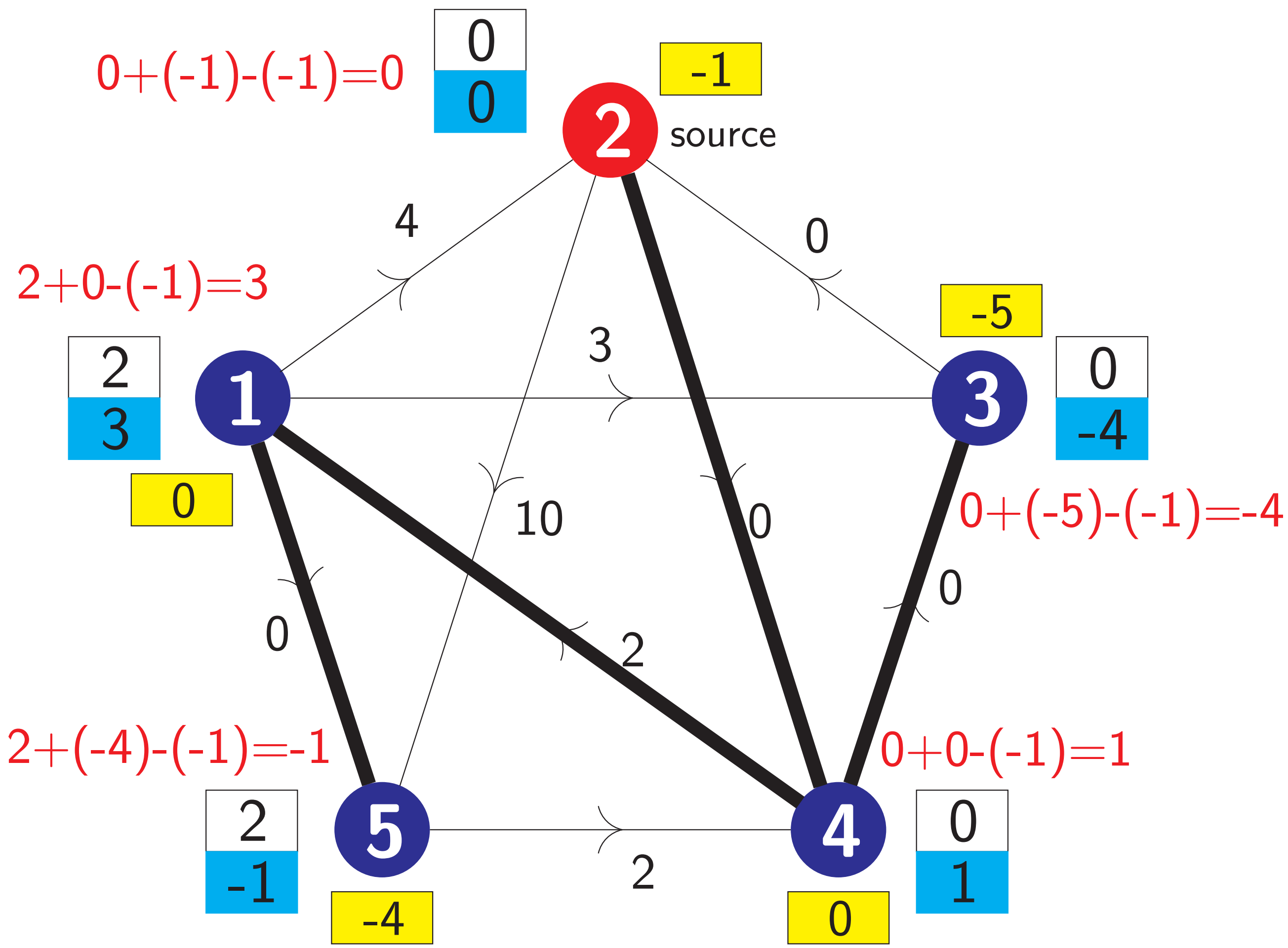
Solution — Running DIJKSTRA'S on vertex 1 as source

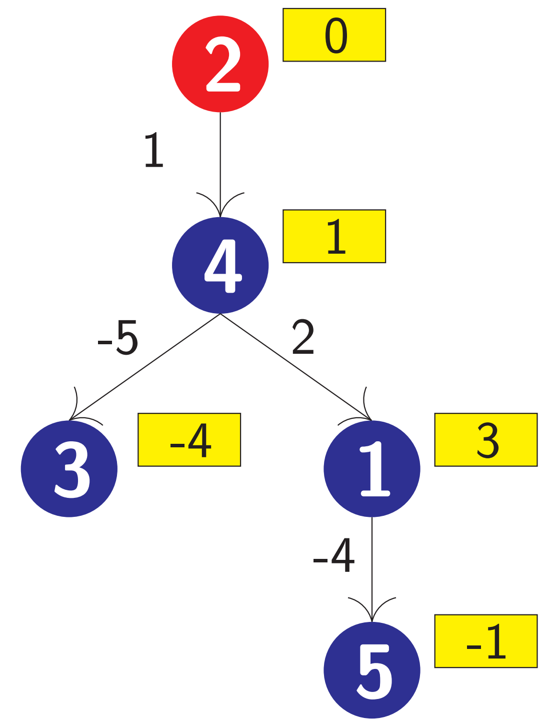


$h[v]$	$\delta(u,\, v)$	$\hat{\delta}(u,\, v) = \delta(u,\, v)+h[v]\text{-}h[u]$

The Johnson's Algorithm

Solution — Running DIJKSTRA'S on vertex 2 as source

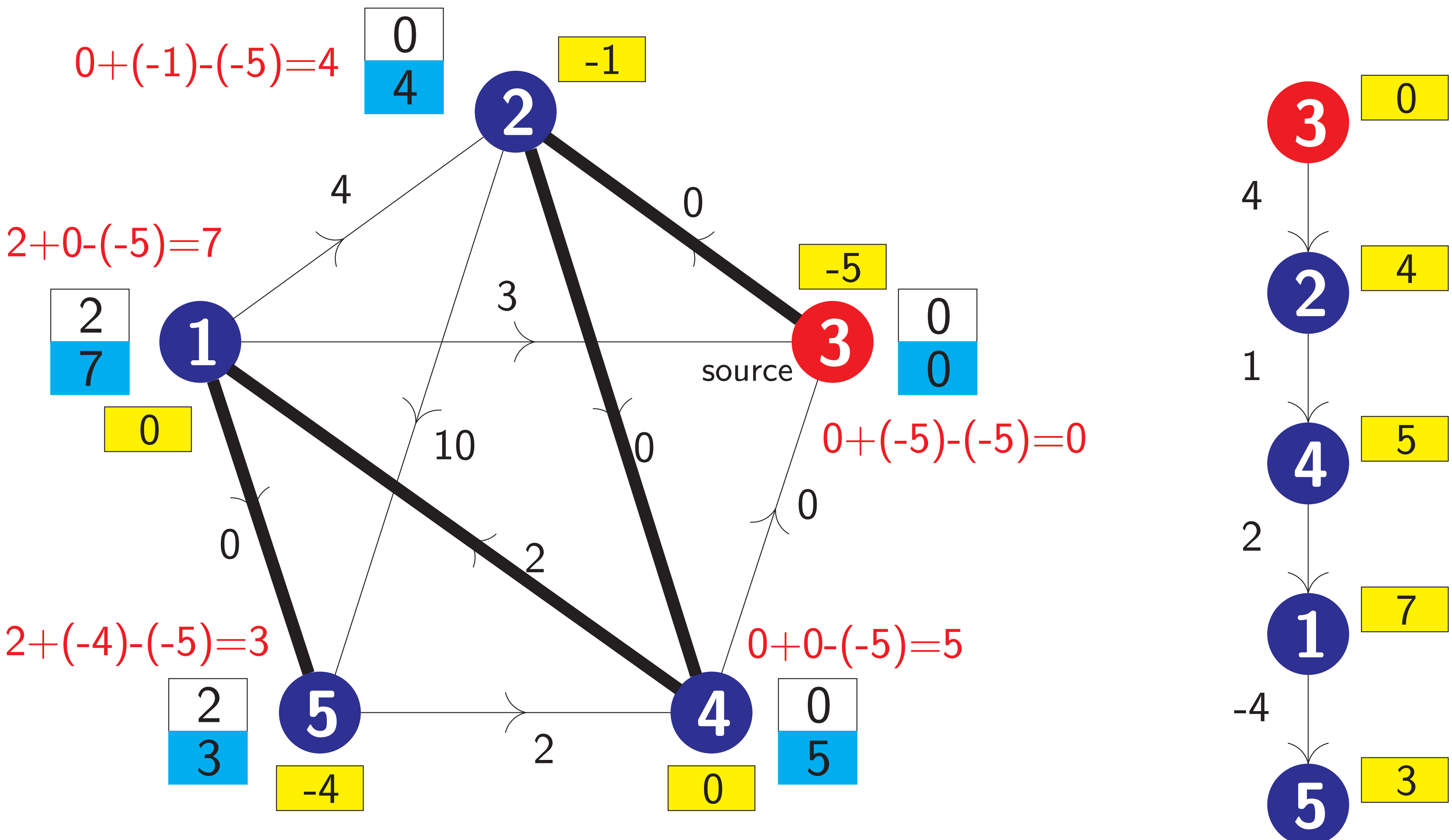




$h[v]$	$\delta(u,\, v)$	$\hat{\delta}(u,\, v) = \delta(u,\, v) + h[v] - h[u]$

The Johnson's Algorithm

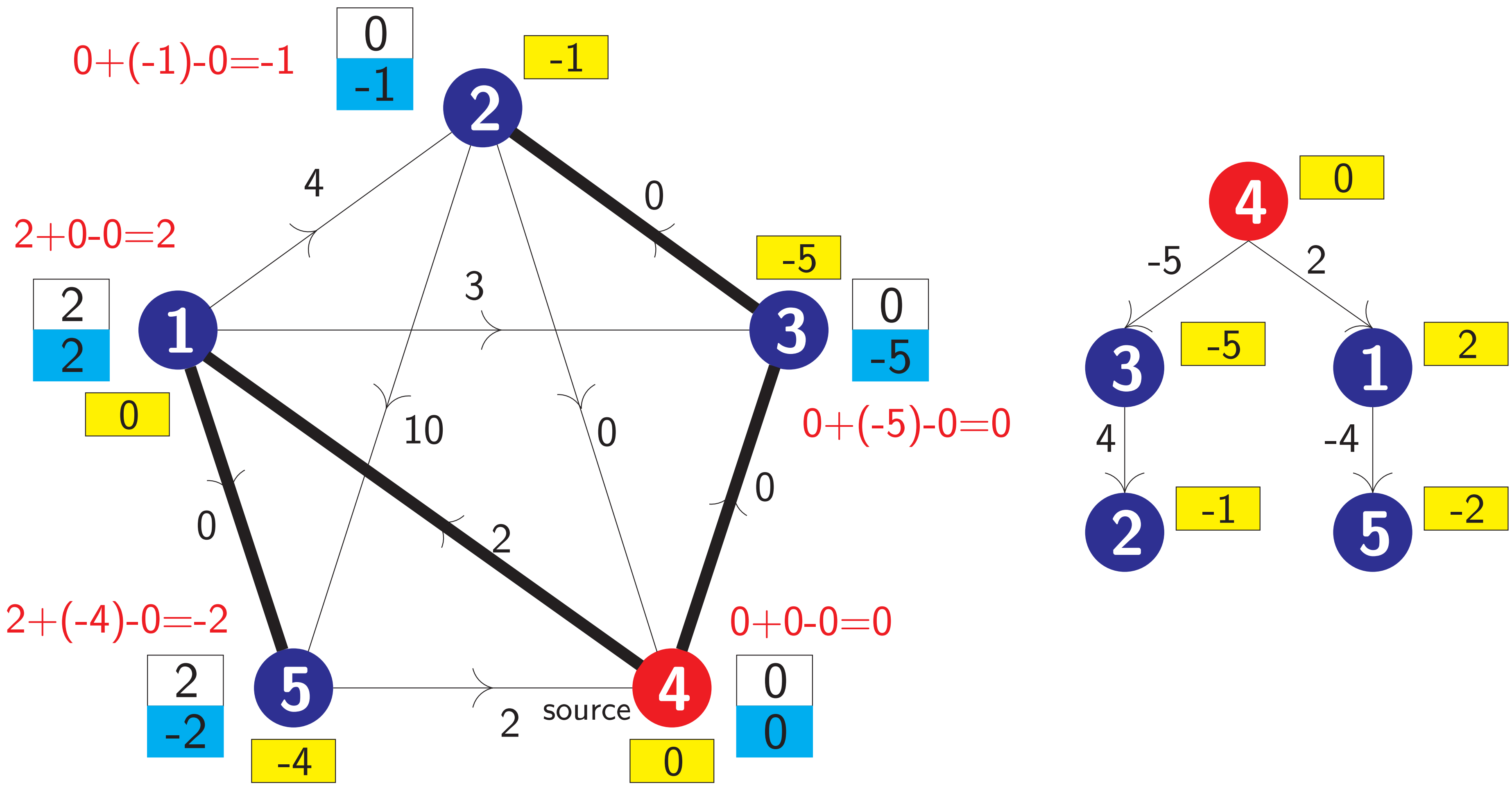
Solution — Running DIJKSTRA'S on vertex 3 as source



$h[v]$	$\delta(u, v)$	$\hat{\delta}(u, v) = \delta(u, v) + h[v] - h[u]$

The Johnson's Algorithm

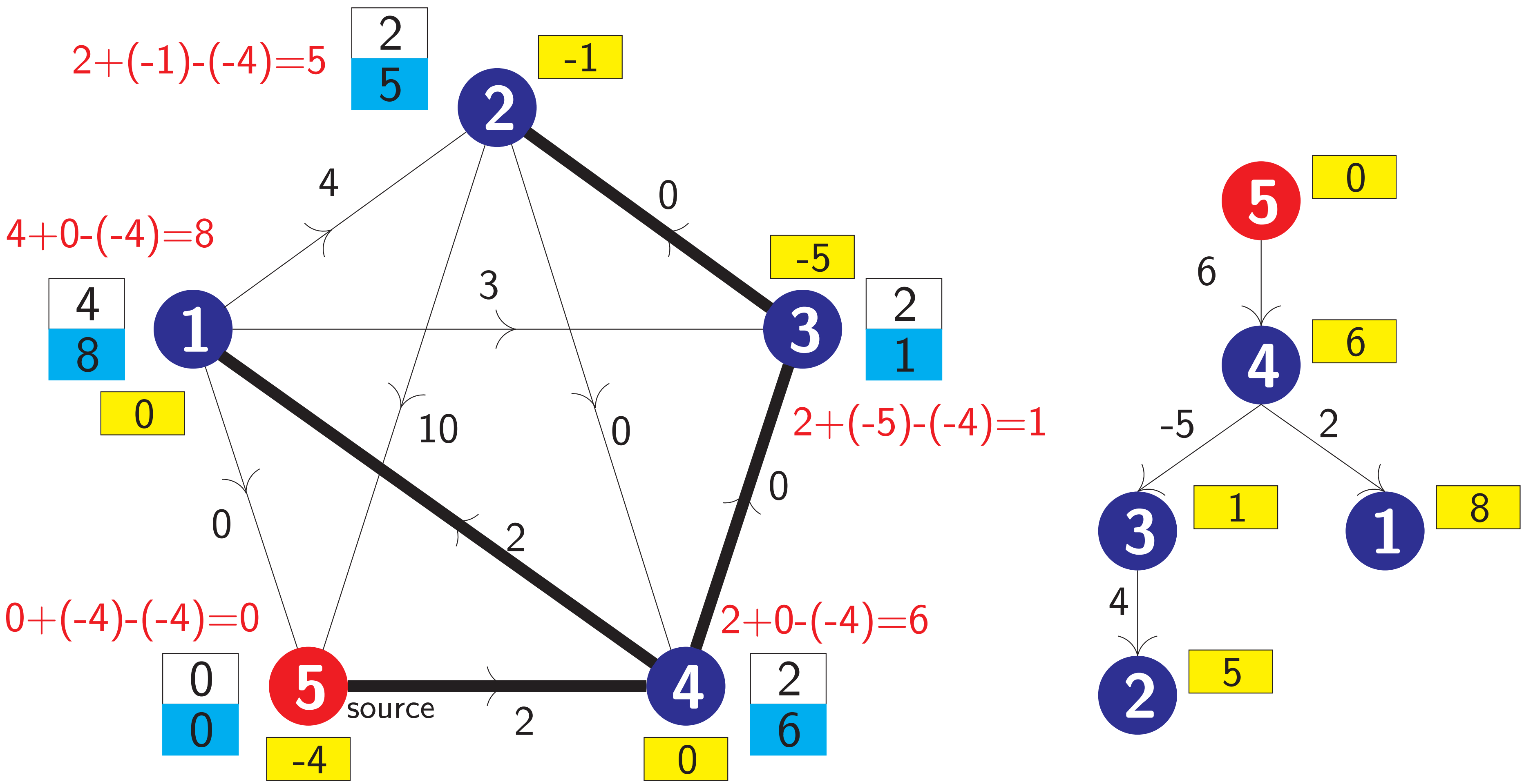
Solution — Running DIJKSTRA'S on vertex 4 as source



$h[v]$	$\delta(u,\, v)$	$\hat{\delta}(u,\, v) = \delta(u,\, v)+h[v]\text{-}h[u]$

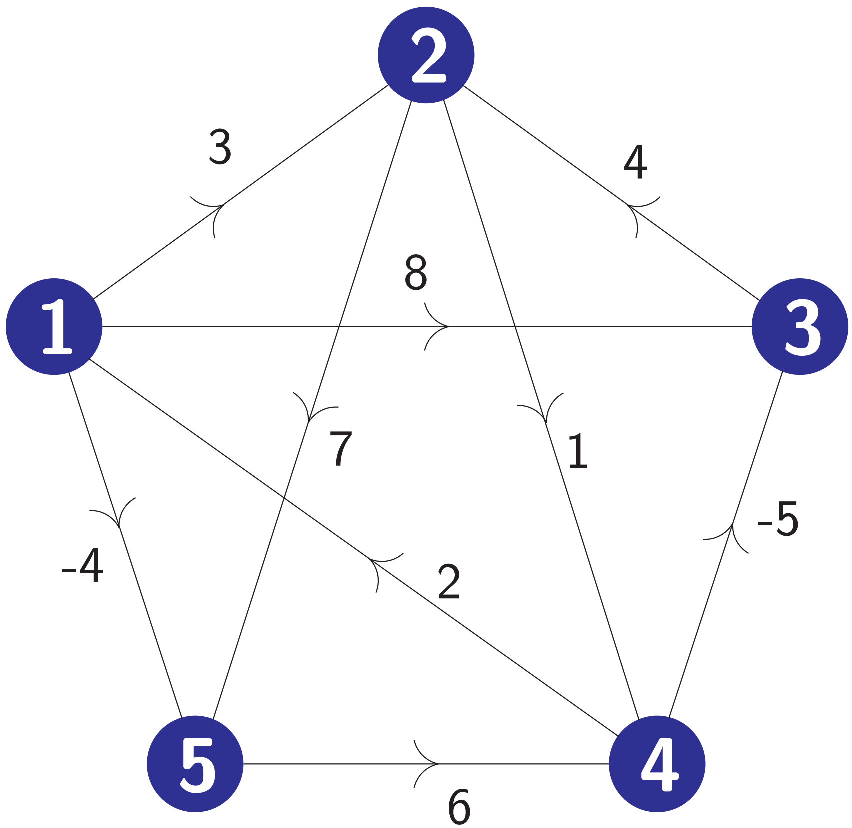
The Johnson's Algorithm

Solution — Running DIJKSTRA'S on vertex 5 as source



$h[v]$	$\delta(u,\, v)$	$\hat{\delta}(u,\, v) = \delta(u,\, v) + h[v] - h[u]$

The Johnson's Algorithm
Solution — Summary



	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0