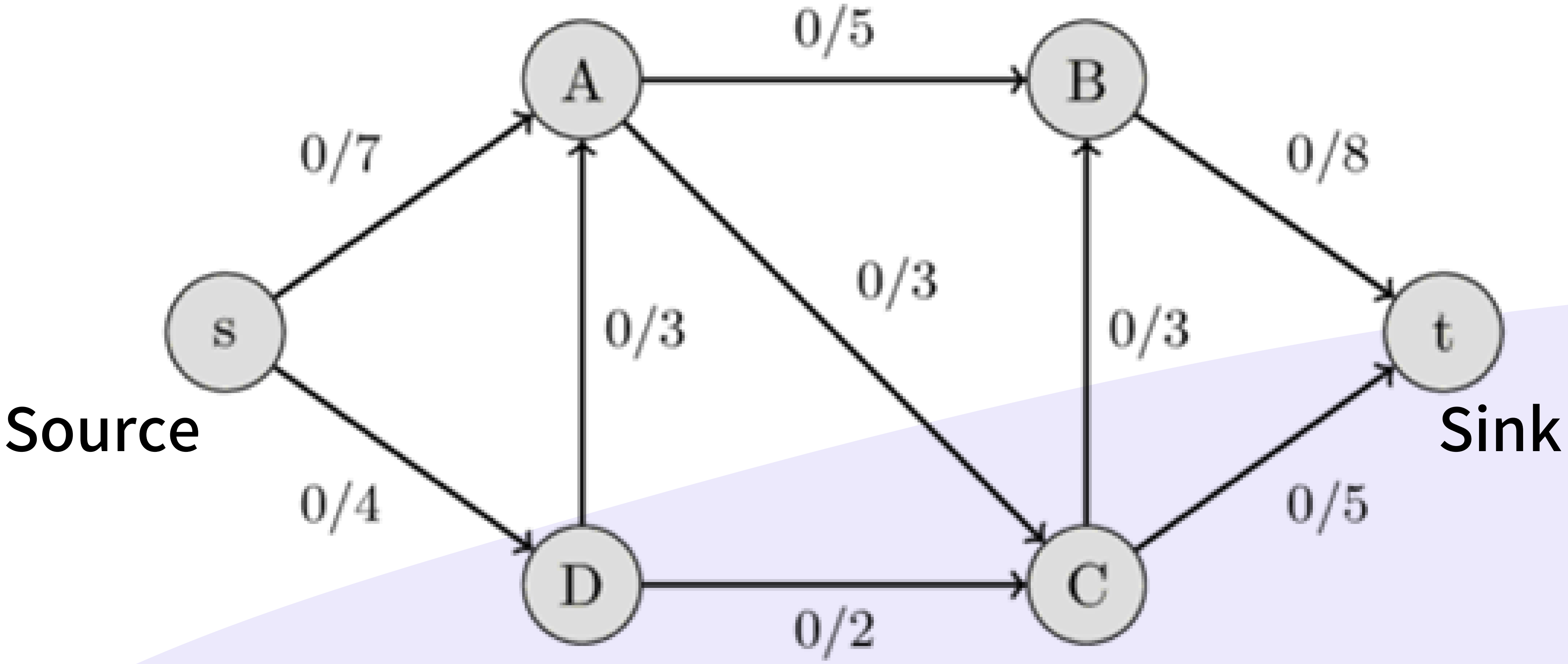
CS101 Discrete Mathematics

Maximum Flow Problem

Dhananjay Goel - 2021CSB1165
Hardik Aggarwal - 2021CSB1173

Problem Description

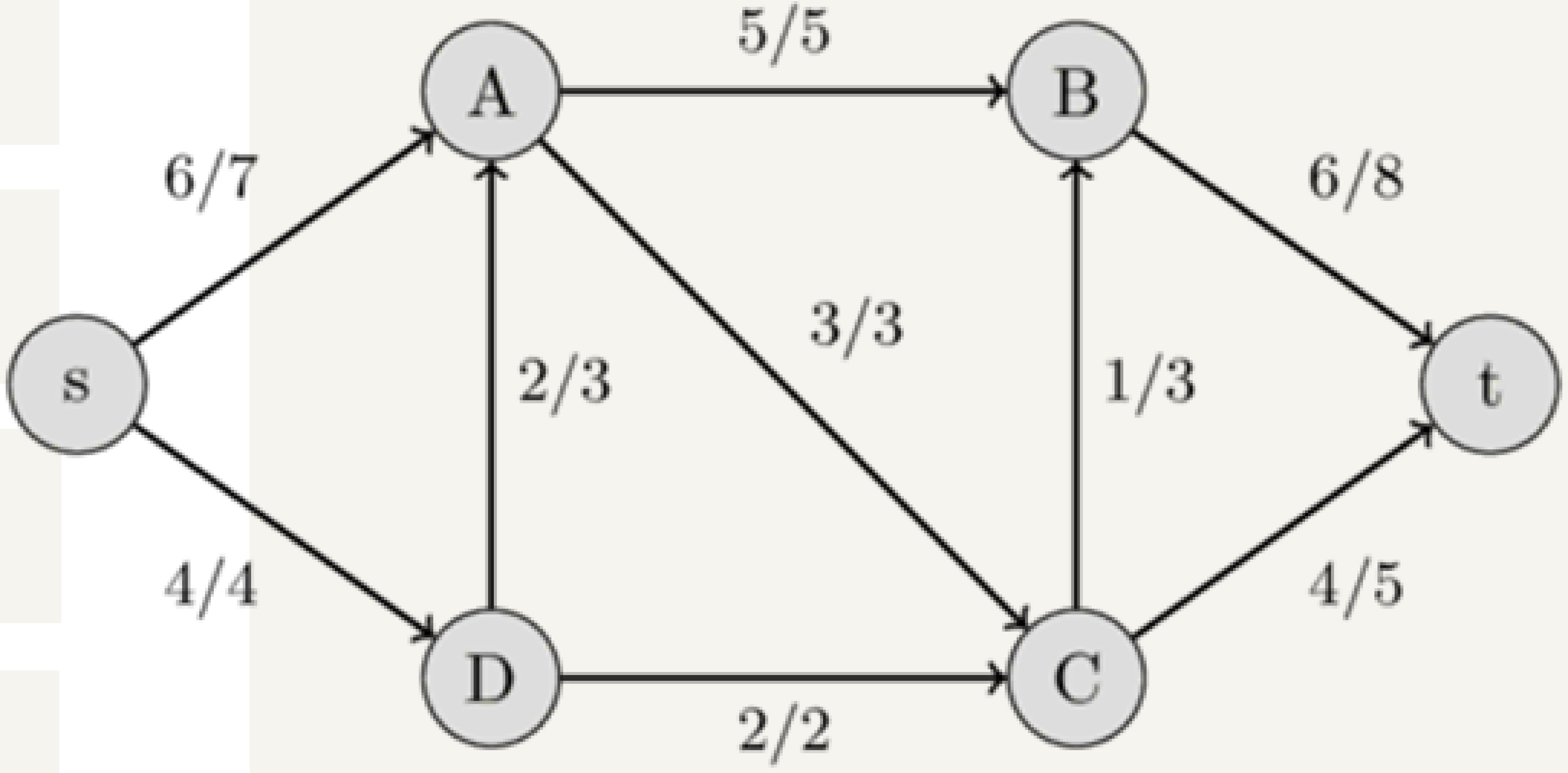
The problem involves determining the most amount of flow that can be sent via a system of pipes, channels and other paths while taking into account the capacity restrictions.



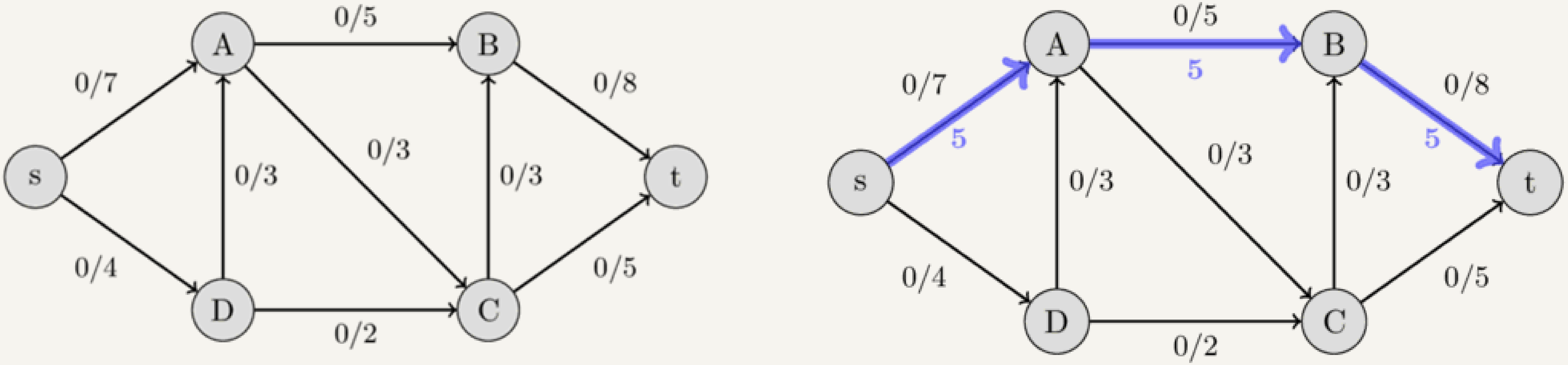
The Directed edge from S to A has maximum capacity of 7 and 0 depicts the amount of flow from the edge. There can be numerous paths to follow to reach sink from source.

How to find the Maximal Flow?

1. Ford-Fulkerson's Algorithm
2. Edmonds-Karp's Algorithm
3. Dinic's Algorithm
4. Capacity Scaling Heuristic

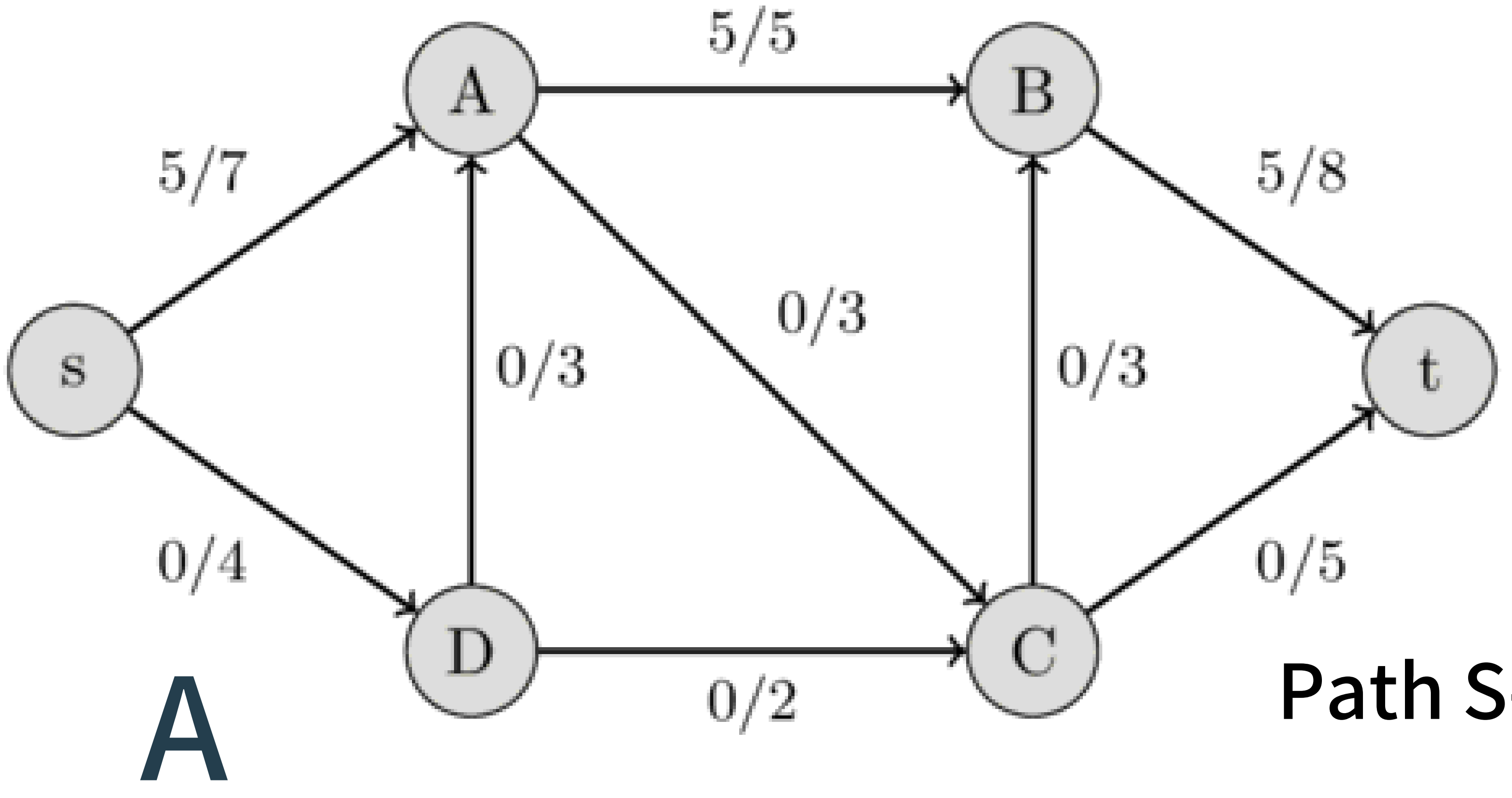


Ford–Fulkerson Algorithm

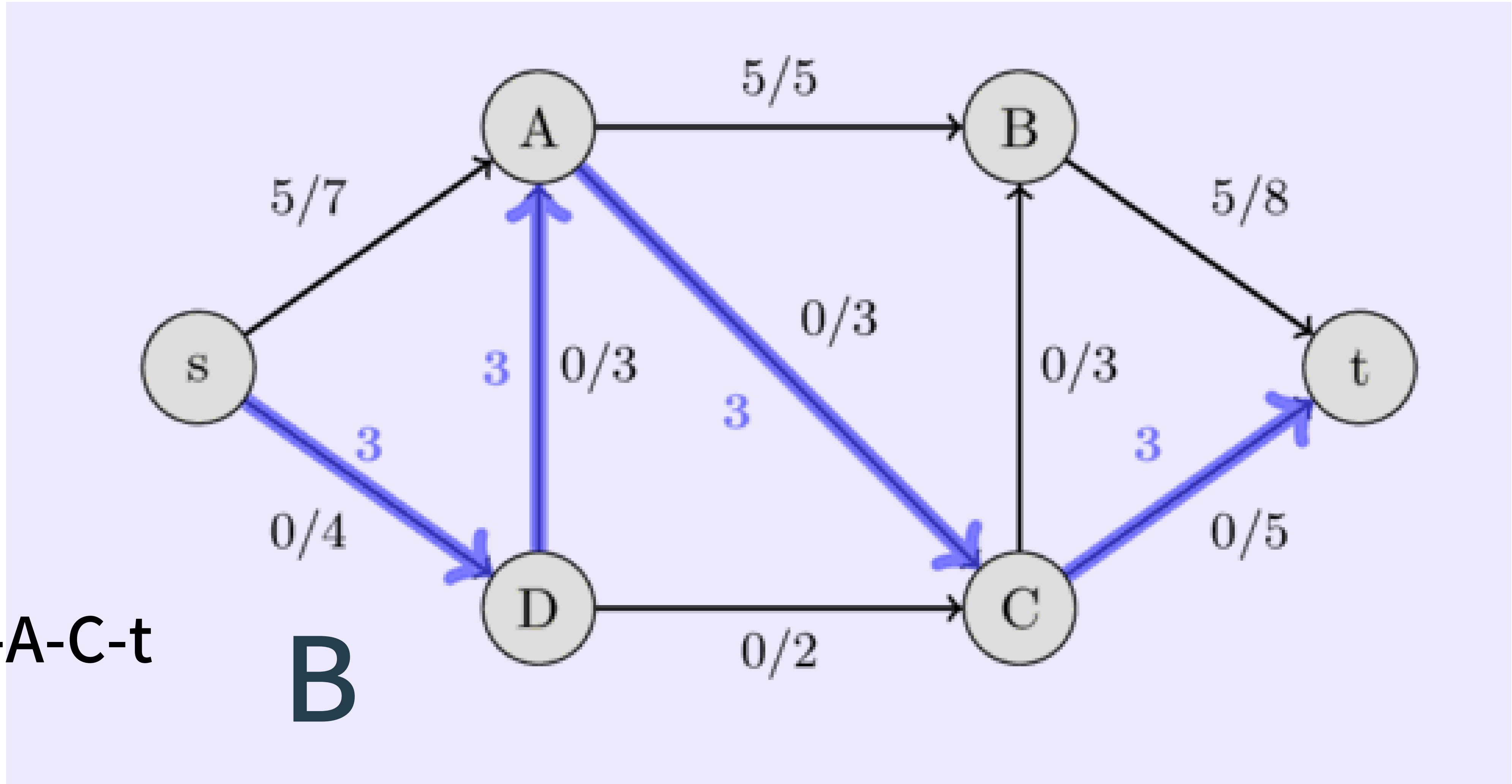


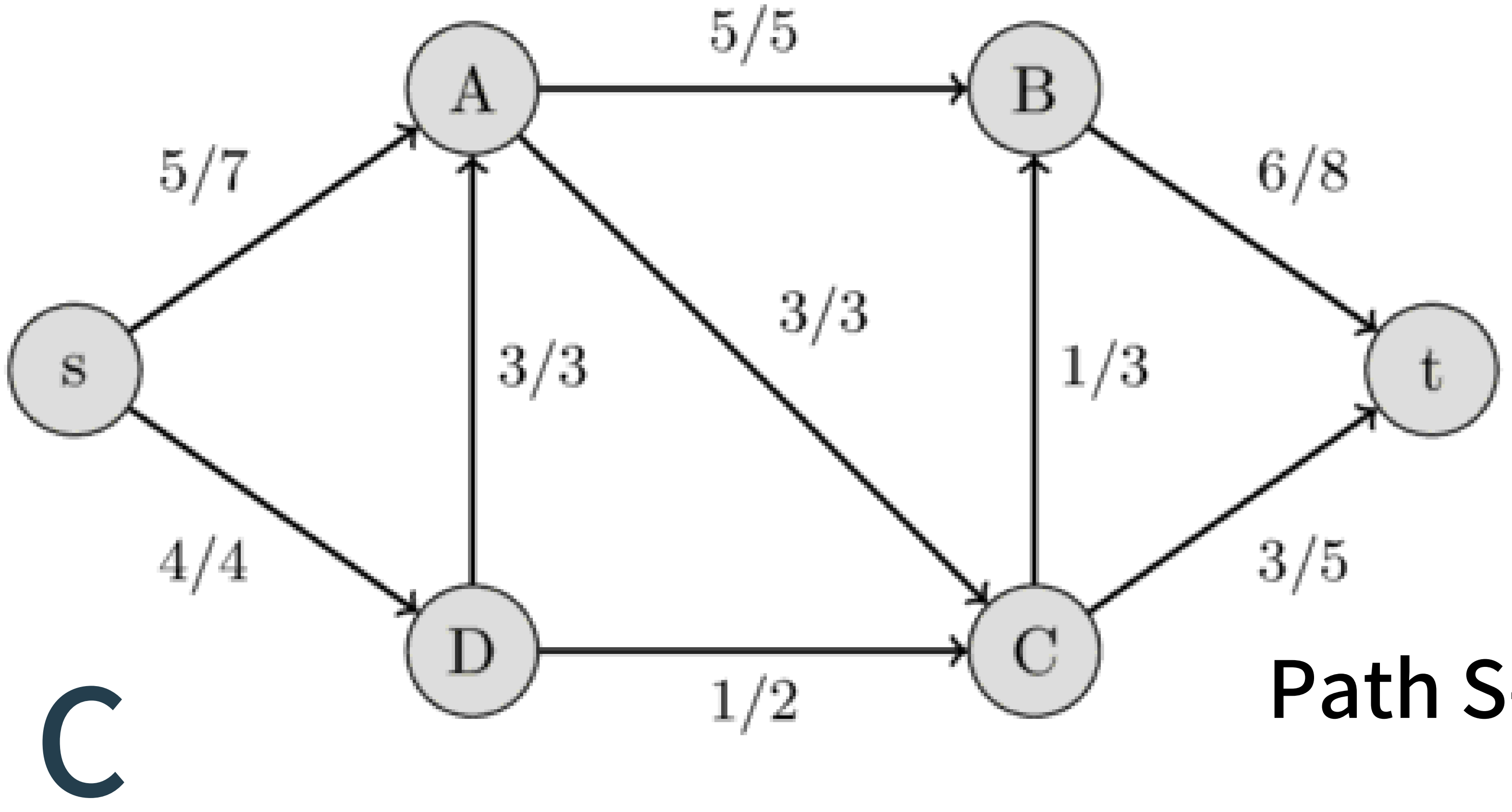
Path S-A-B-t

Uses DFS to find augmenting paths

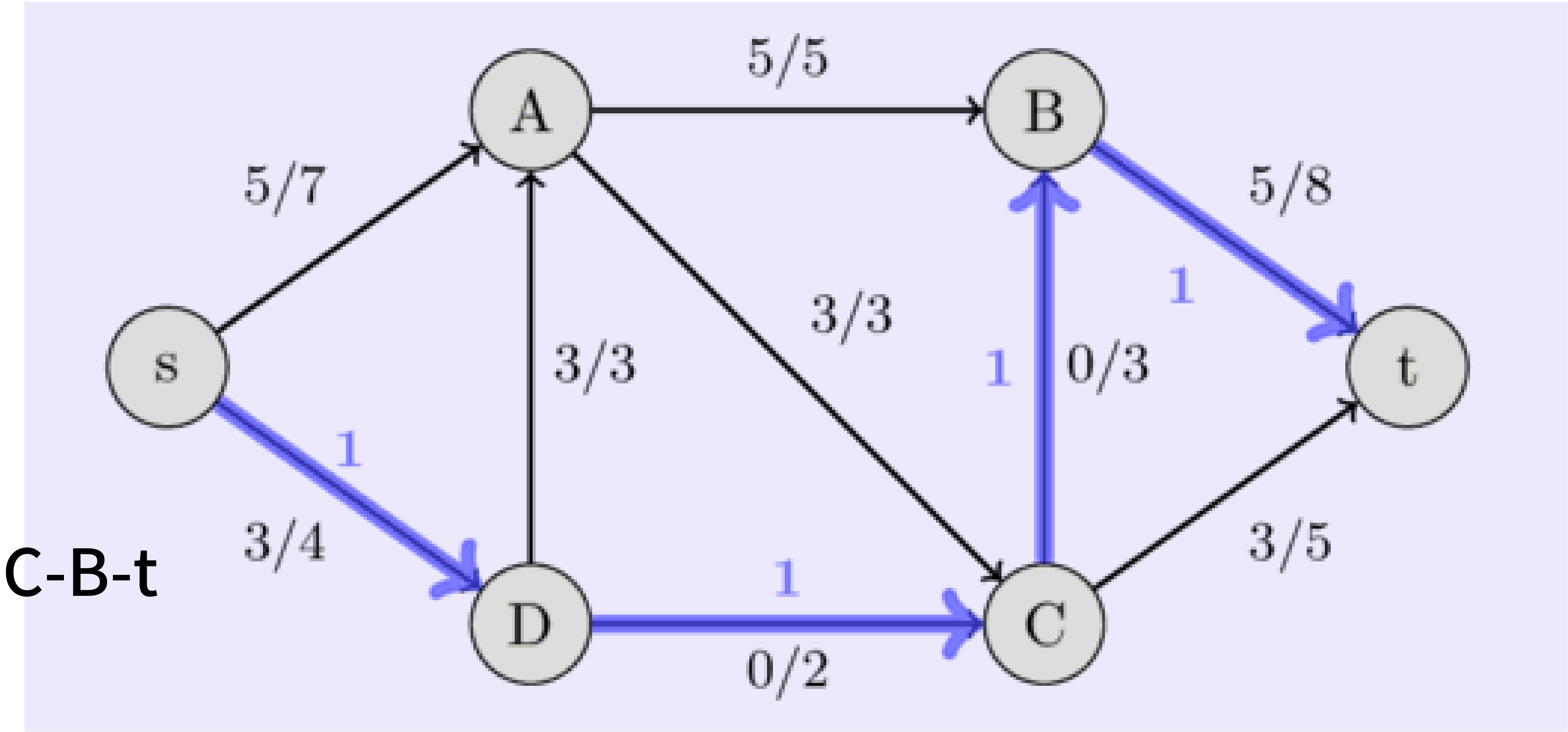


Path S-D-A-C-t

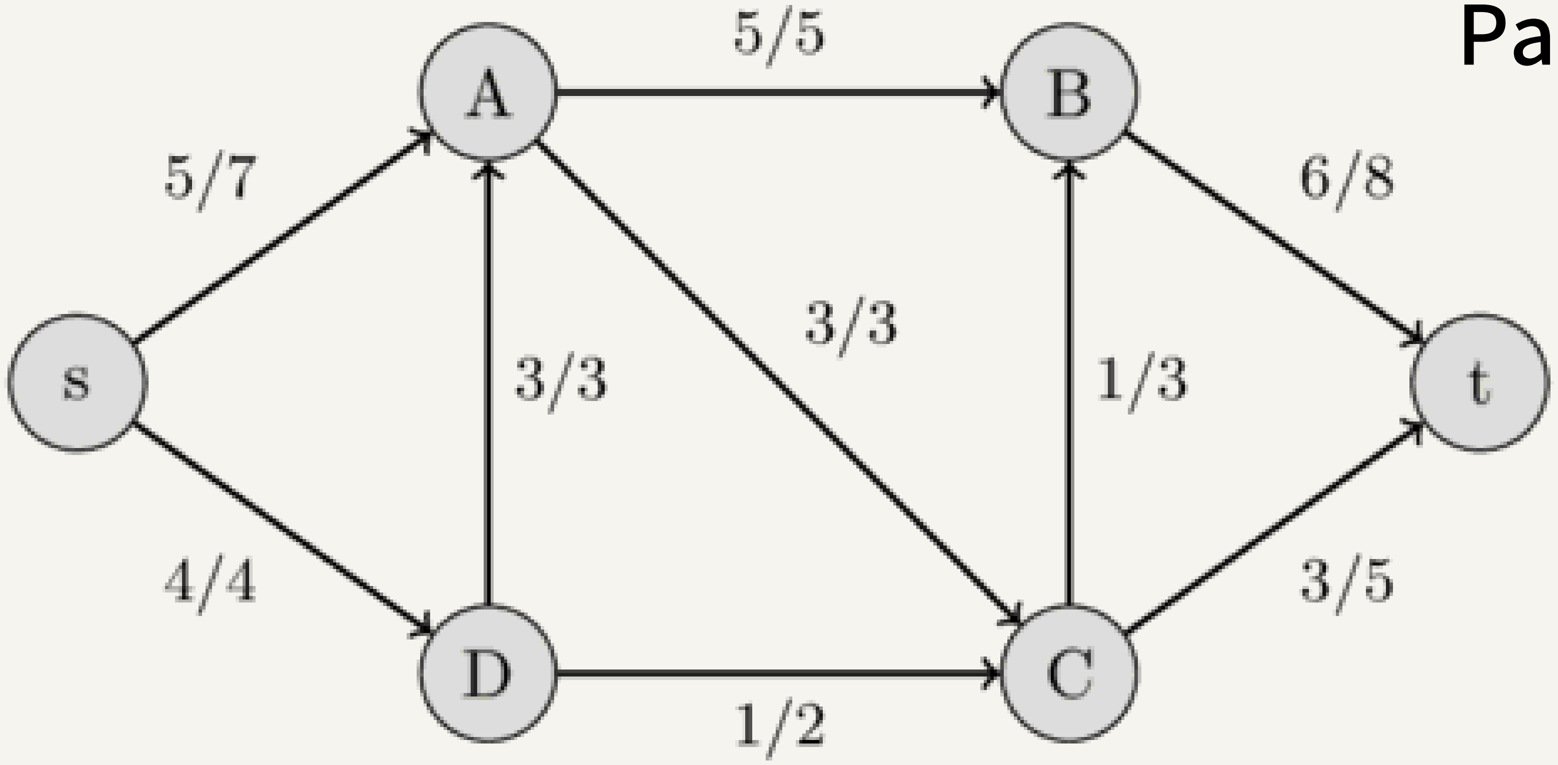




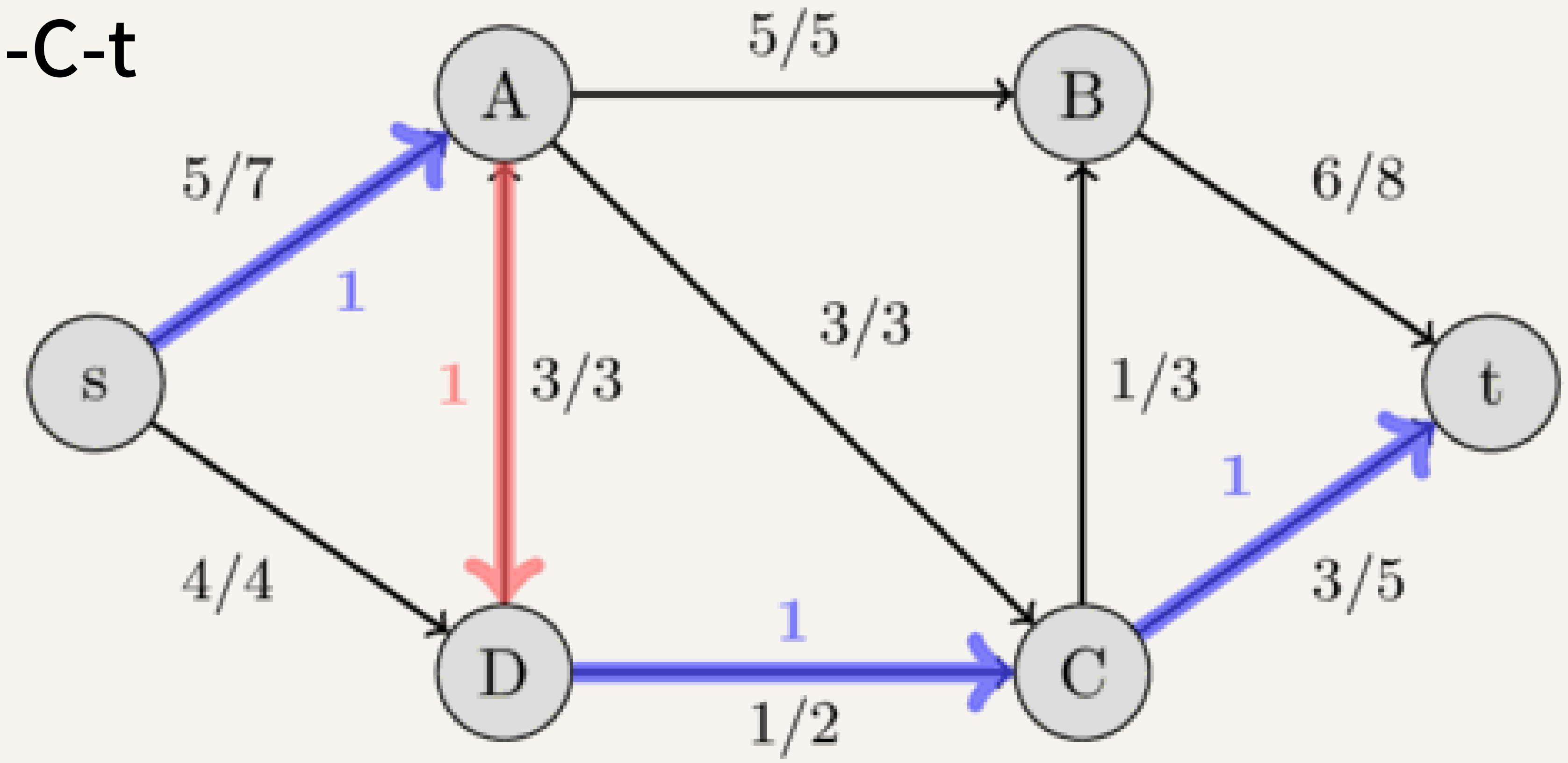
Path S-D-C-B-t



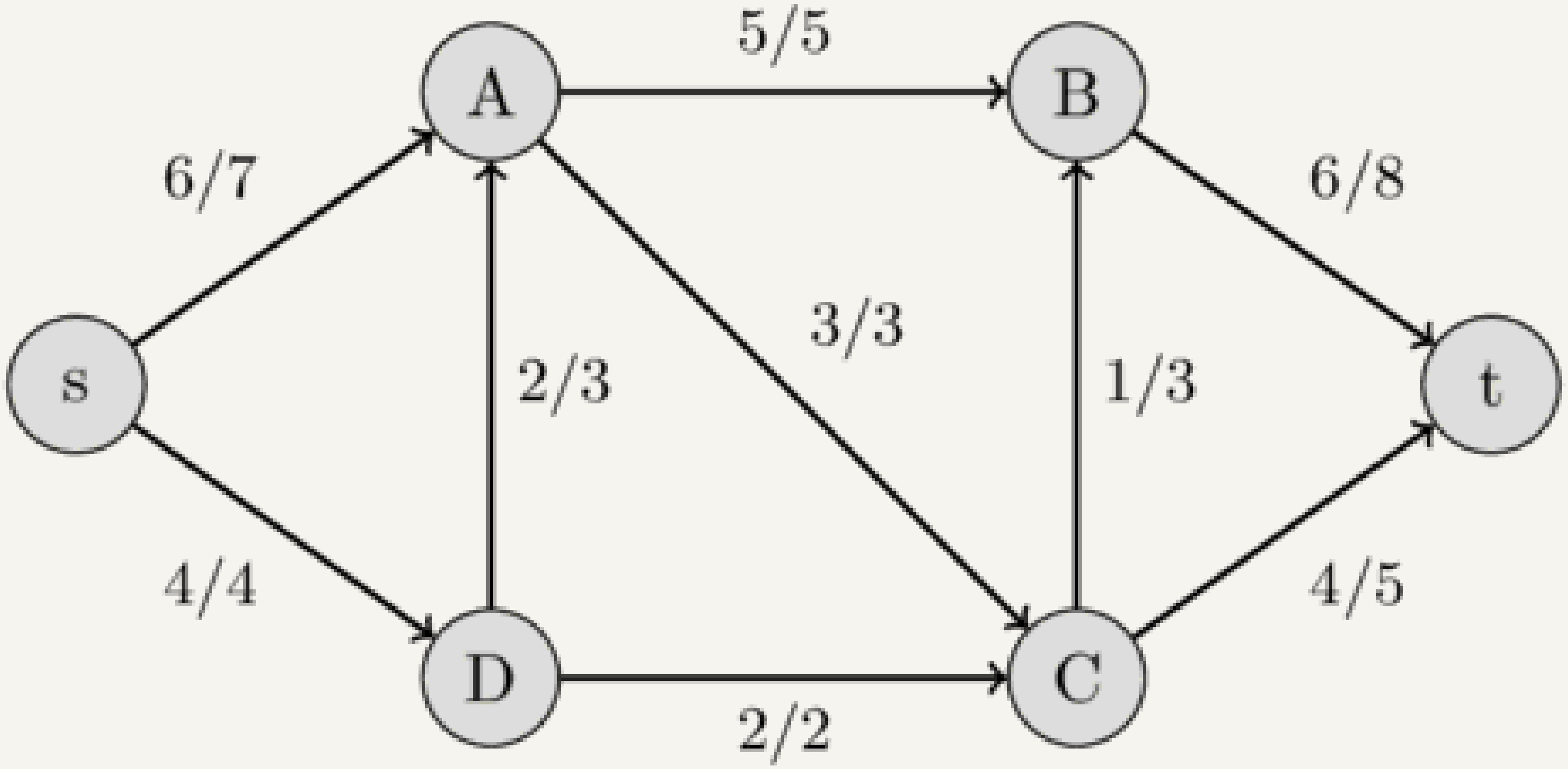
Path S-A-D-C-t



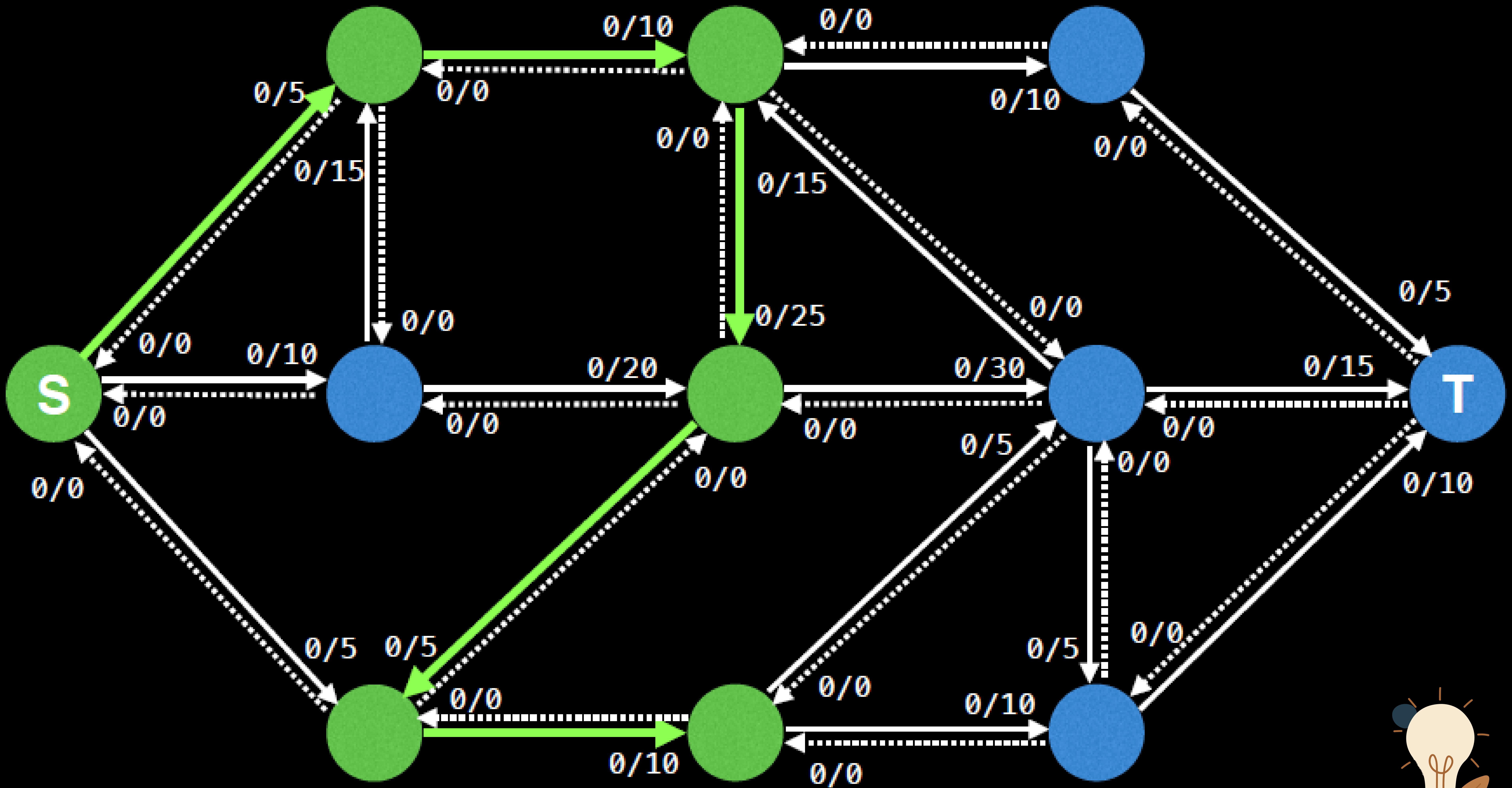
A
B
s
t
D
C
5/5
5/7
6/8
3/3
3/3
1/3
4/4
3/5
1/2



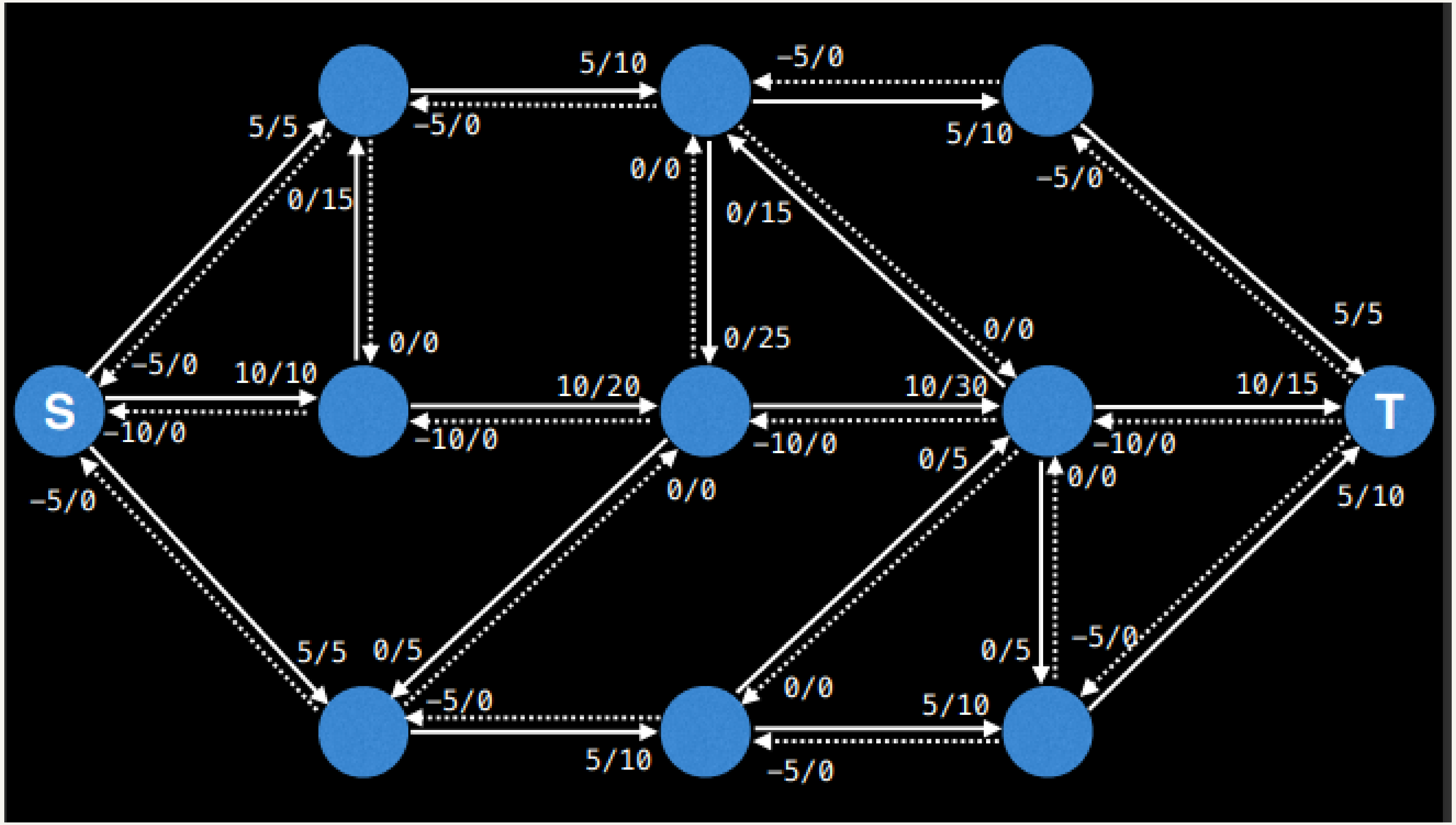
A
B
s
t
D
C
5/5
5/7
1
6/8
1
3/3
3/3
1/3
1
4/4
1
3/5
1/2

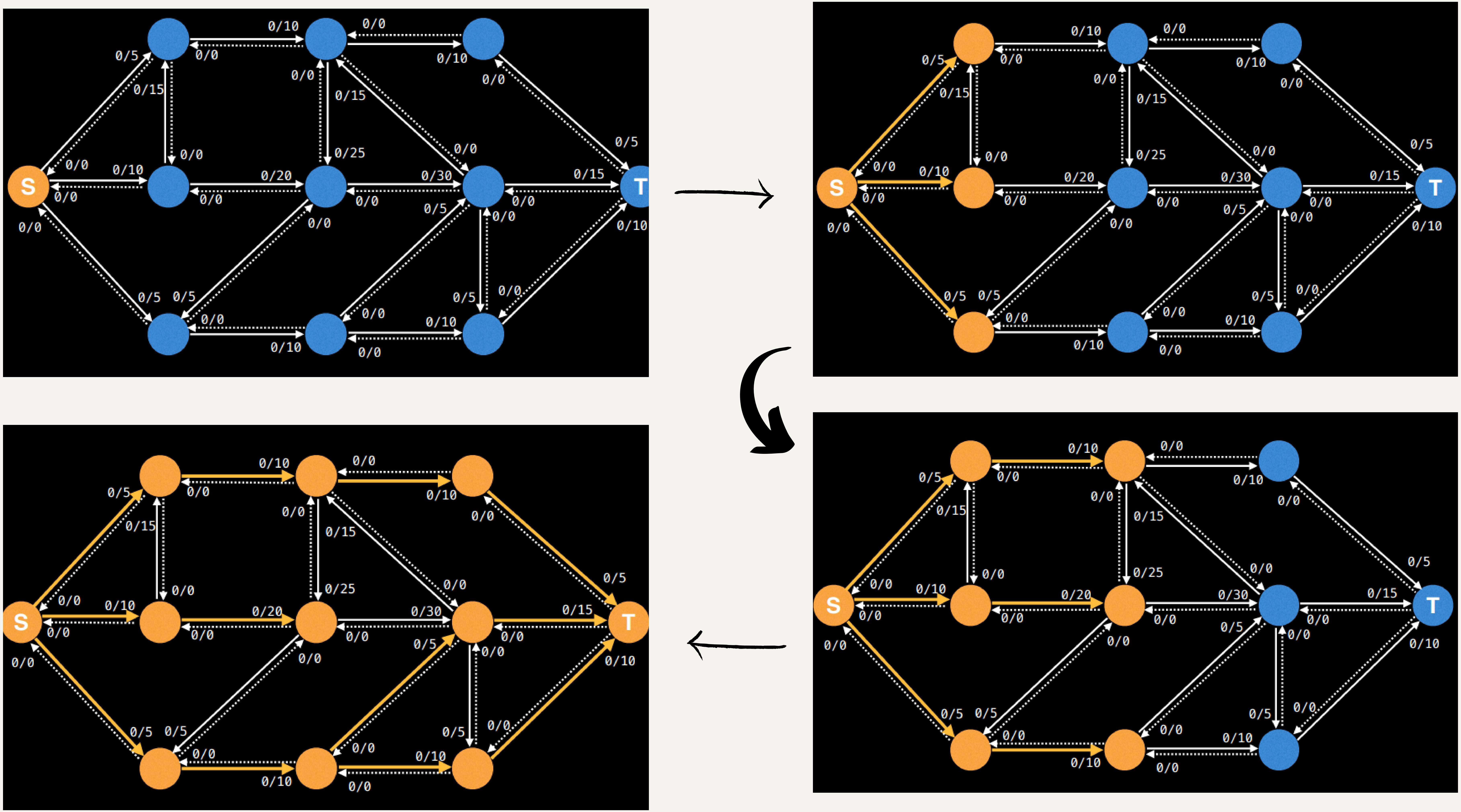


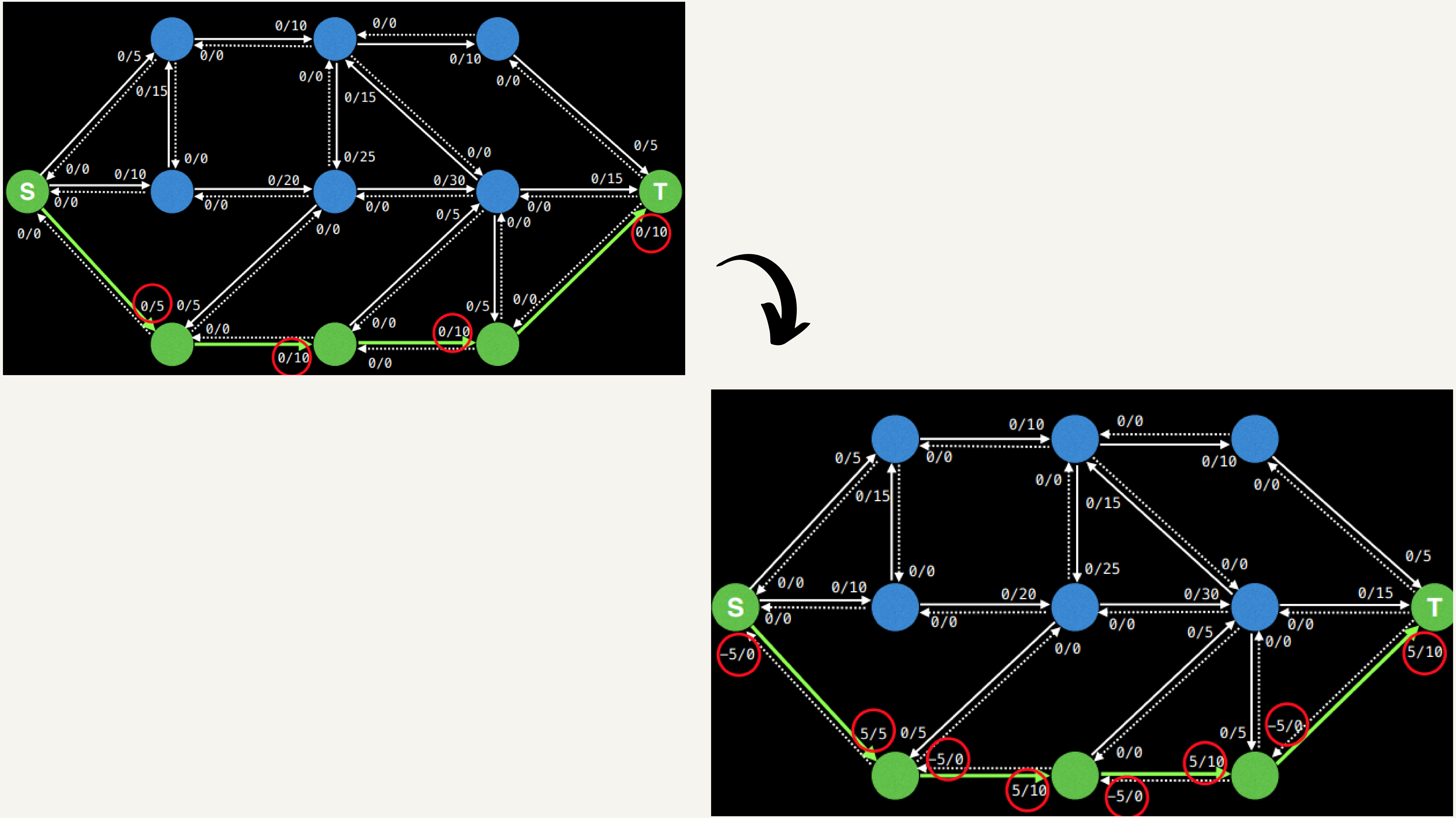
A
B
s
t
D
C
5/5
6/7
6/8
2/3
3/3
1/3
4/4
4/5
2/2


0/10
0/0
0/5
0/0
0/10
0/0
0/15
0/0
0/15
0/5
0/0
0/25
0/0
0/0
0/10
0/20
0/30
0/15
S
T
0/0
0/0
0/0
0/0
0/5
0/0
0/0
0/10
0/5
0/5
0/0
0/5
0/0
0/0
0/10
0/10
0/0

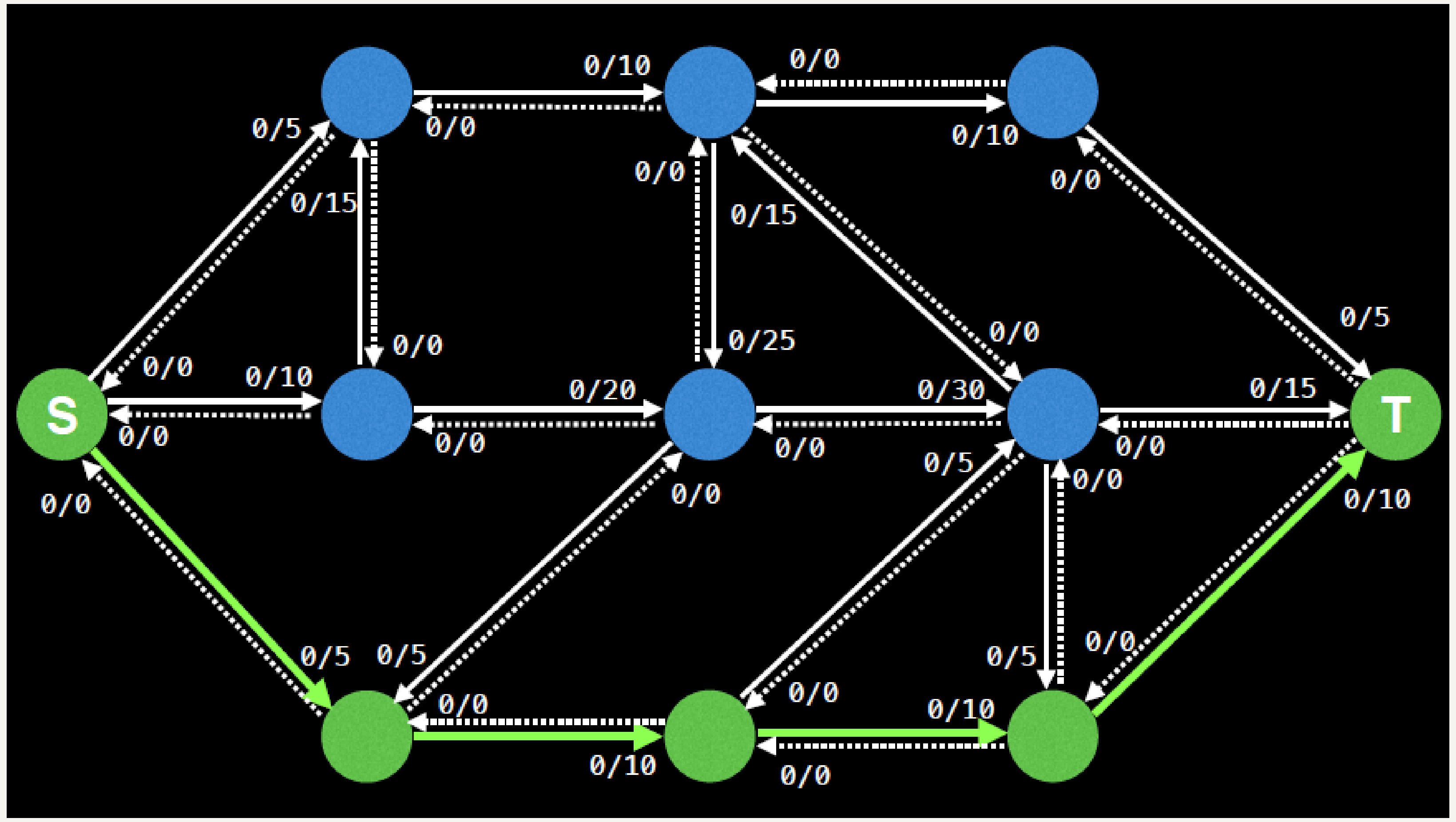
Edmonds Karp Algorithm

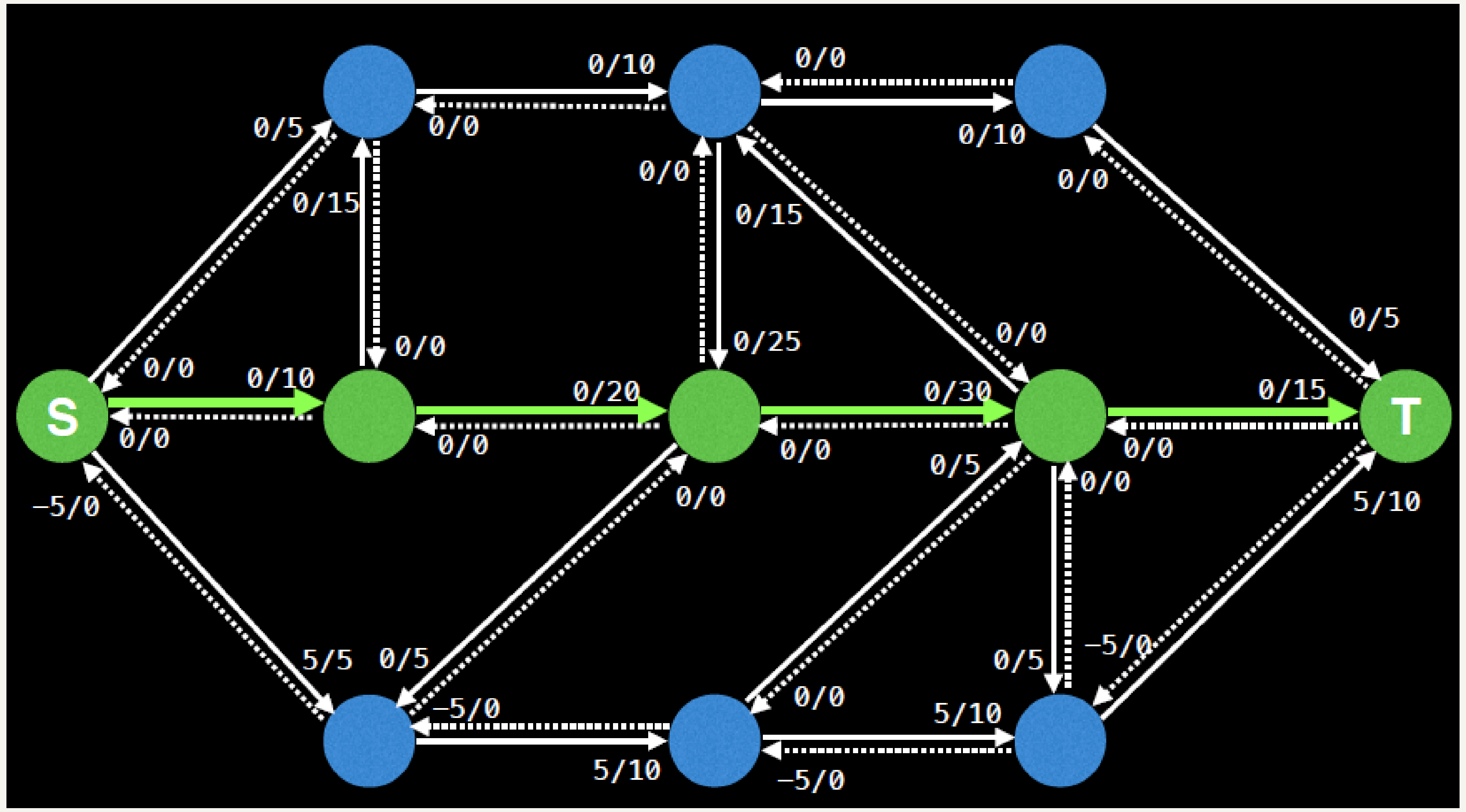


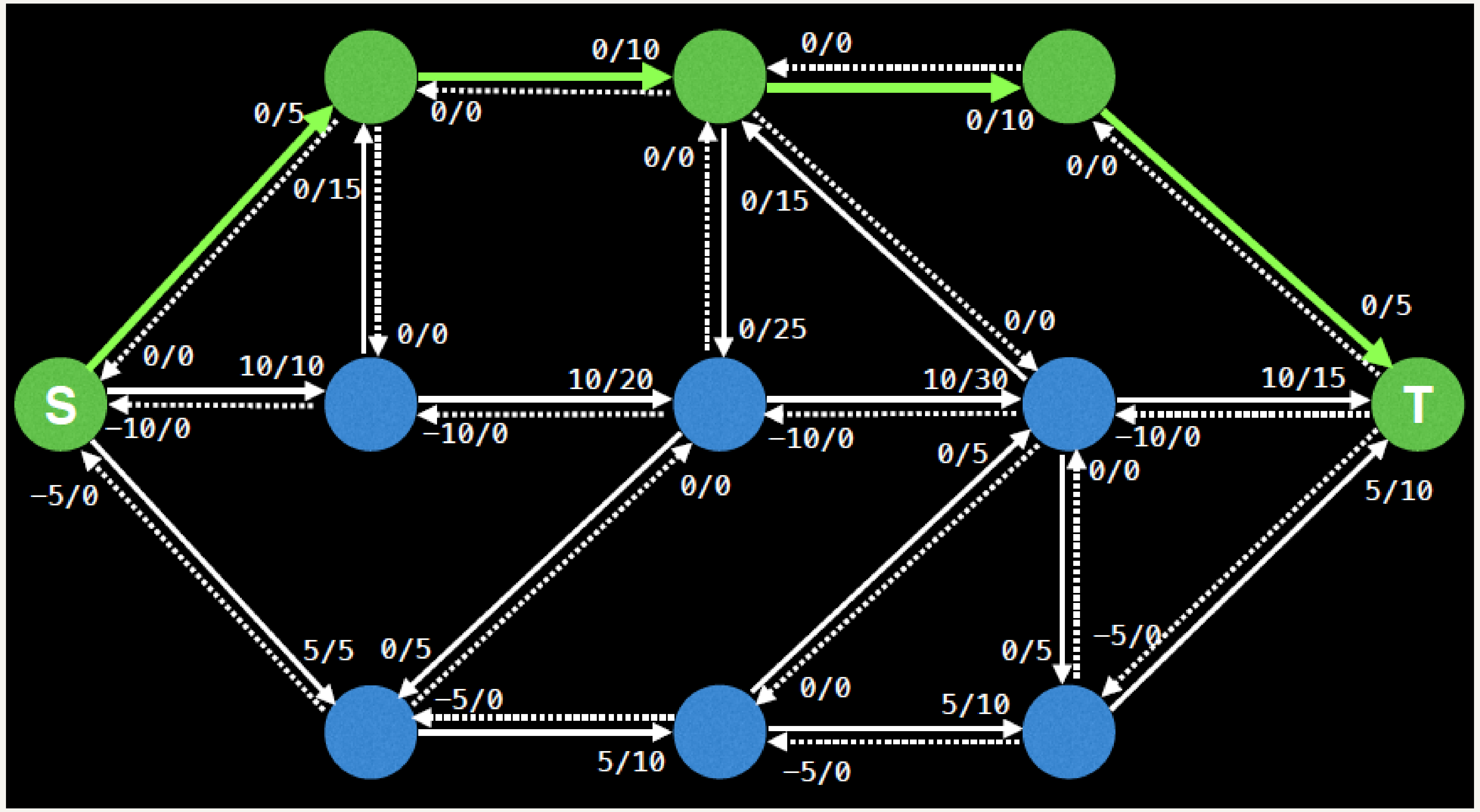

S
T
0/5
0/10
0/0
0/15
0/25
0/20
0/30
0/15
0/5
0/10
S
T
0/5
0/10
0/0
0/15
0/25
0/20
0/30
0/15
0/5
0/10
S
T
0/5
0/10
0/0
0/15
0/25
0/20
0/30
0/15
0/5
0/10
S
T
0/5
0/10
0/0
0/15
0/25
0/20
0/30
0/15
0/5
0/10



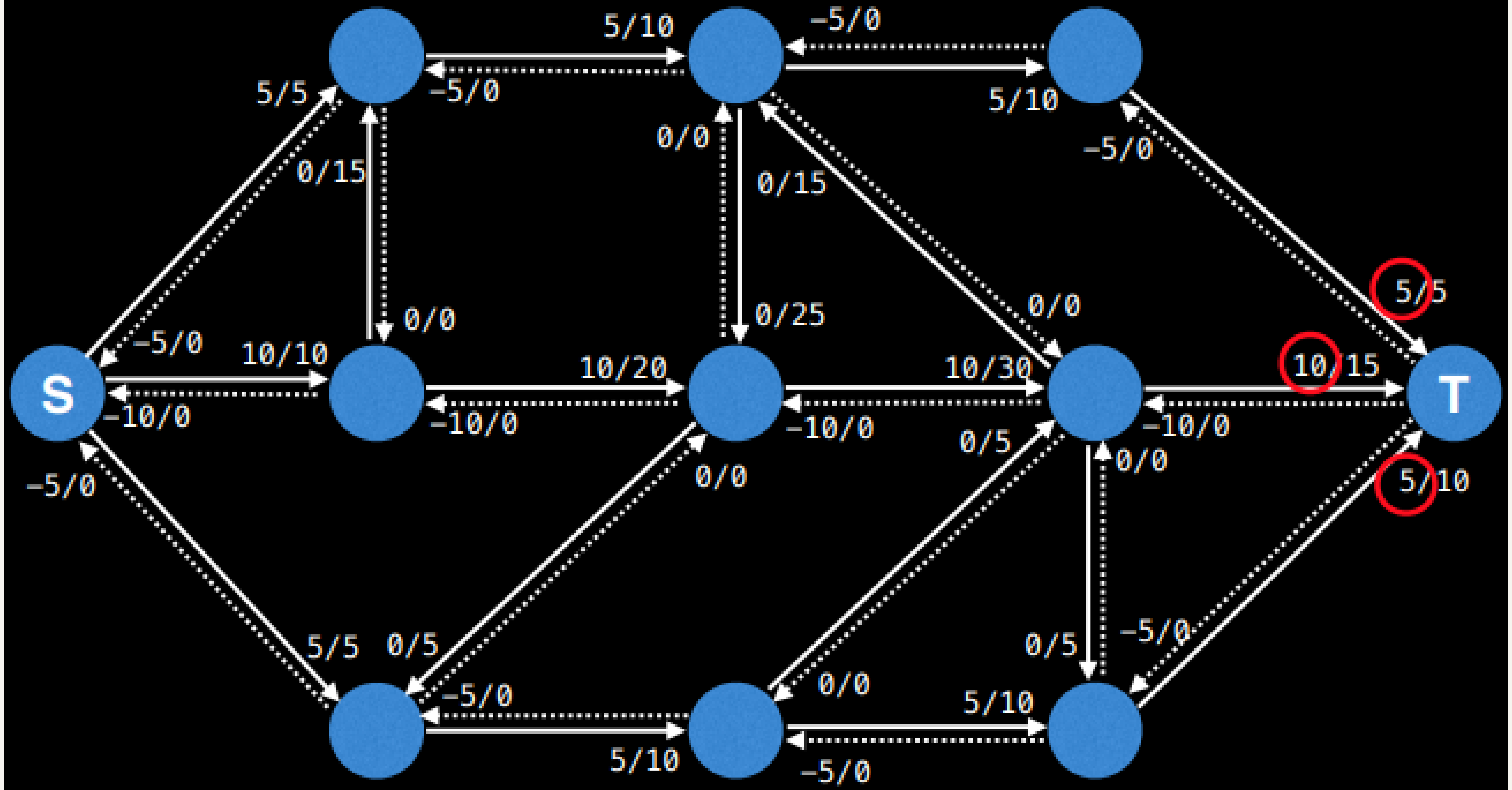
S
T
0/5
0/10
0/0
0/0
0/10
0/15
0/0
0/15
0/0
0/0
0/25
0/0
0/5
0/0
0/10
0/0
0/20
0/30
0/15
0/0
0/0
0/0
0/0
0/5
0/0
0/10
0/0
0/5
0/5
0/0
0/0
0/10
0/5
0/0
0/10
0/0
S
T
0/5
0/10
0/0
0/0
0/10
0/15
0/0
0/15
0/0
0/0
0/25
0/0
0/5
0/0
0/10
0/0
0/20
0/30
0/15
0/0
0/0
0/0
0/0
0/5
0/0
-5/0
5/10
5/5
0/5
0/0
0/5
-5/0
-5/0
5/10
5/10
-5/0


S
T
0/10
0/0
0/5
0/0
0/10
0/15
0/0
0/0
0/15
0/5
0/0
0/25
0/0
0/0
0/10
0/20
0/30
0/15
0/0
0/0
0/0
0/0
0/0
0/0
0/5
0/0
0/10
0/0
0/5
0/5
0/0
0/5
0/0
0/0
0/10
0/10
0/0

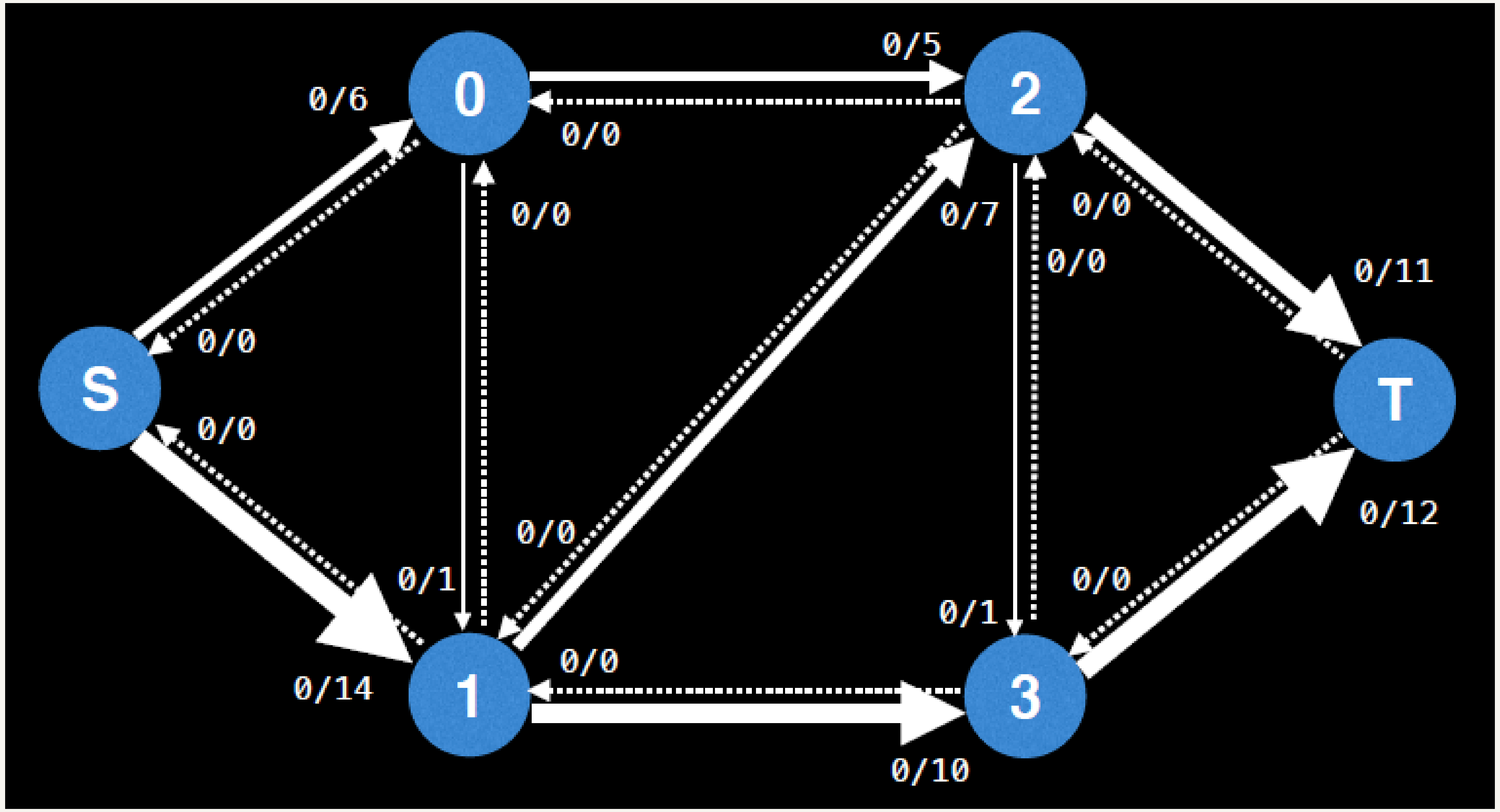

S
T
0/10
0/0
0/5
0/0
0/10
0/15
0/0
0/0
0/15
0/5
0/0
0/25
0/0
0/0
0/10
0/20
0/30
0/15
0/0
0/0
0/0
0/0
-5/0
0/0
0/5
0/0
5/10
5/5
0/5
-5/0
0/5
-5/0
0/0
5/10
5/10
-5/0


S
T
0/10
0/0
0/5
0/0
0/10
0/15
0/0
0/0
0/15
0/5
0/0
0/25
0/0
-10/0
10/10
10/20
10/30
10/15
-10/0
-10/0
-10/0
-5/0
0/0
0/5
0/0
5/10
5/5
0/5
-5/0
0/5
-5/0
0/0
5/10
5/10
-5/0

Max flow = 5 + 10 + 5 = 20



Capacity Scaling Heuristic

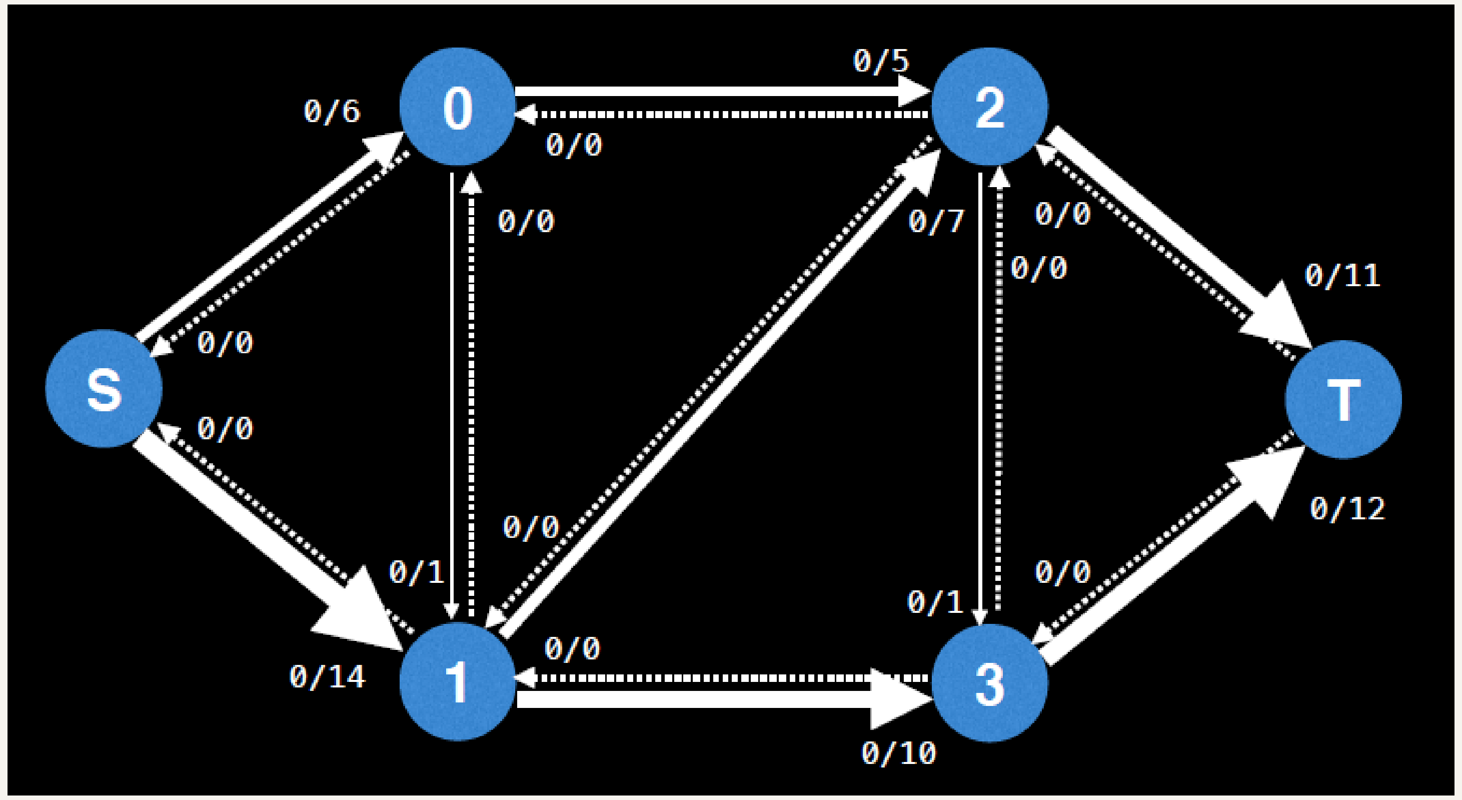


Adjust the size of each edge based on the capacity value highlighting the edges that should be given priority while finding maximal flow

Workflow

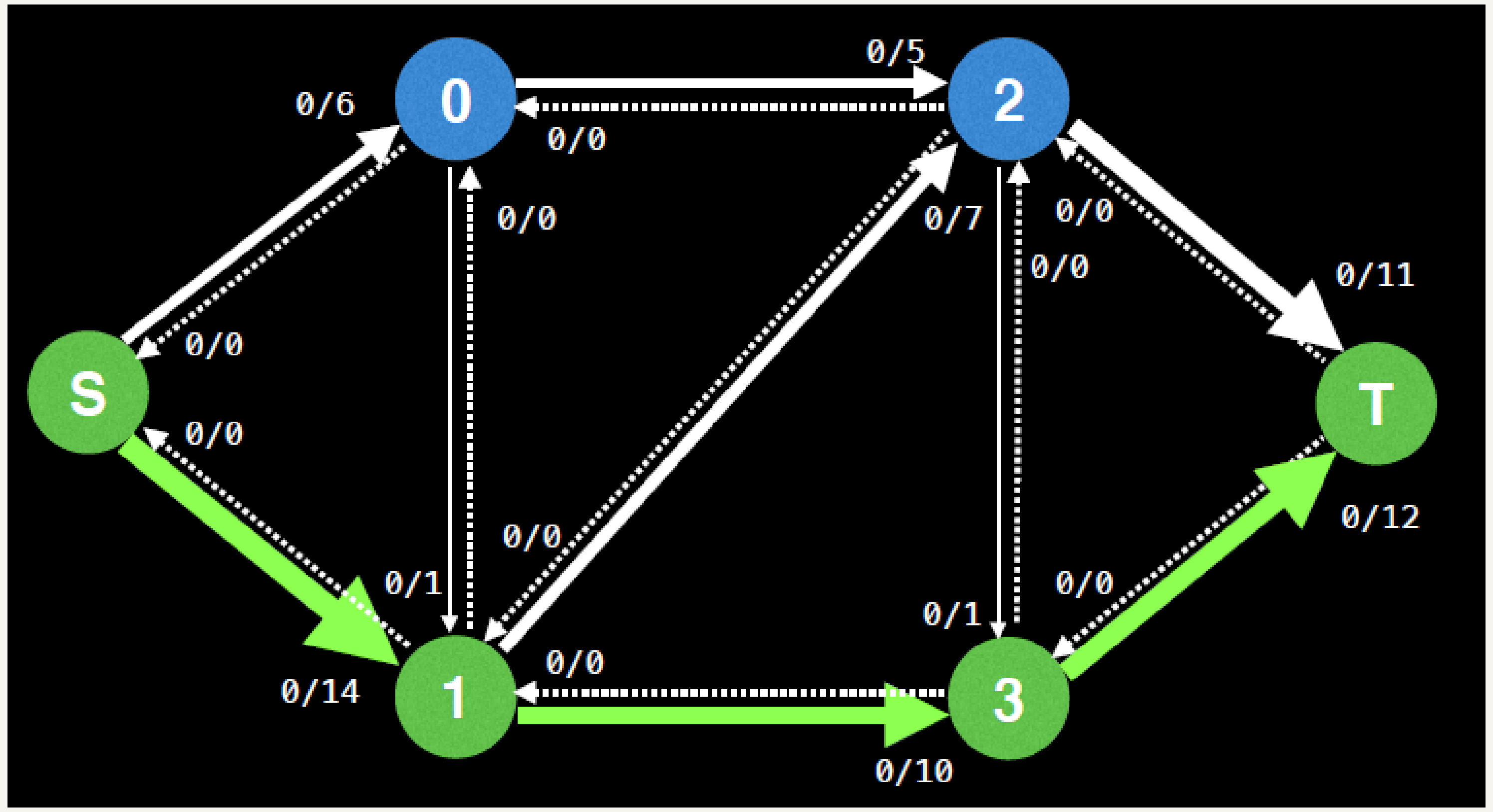
U = largets edge capacity

Δ = largest power of 2 less than or equal to U

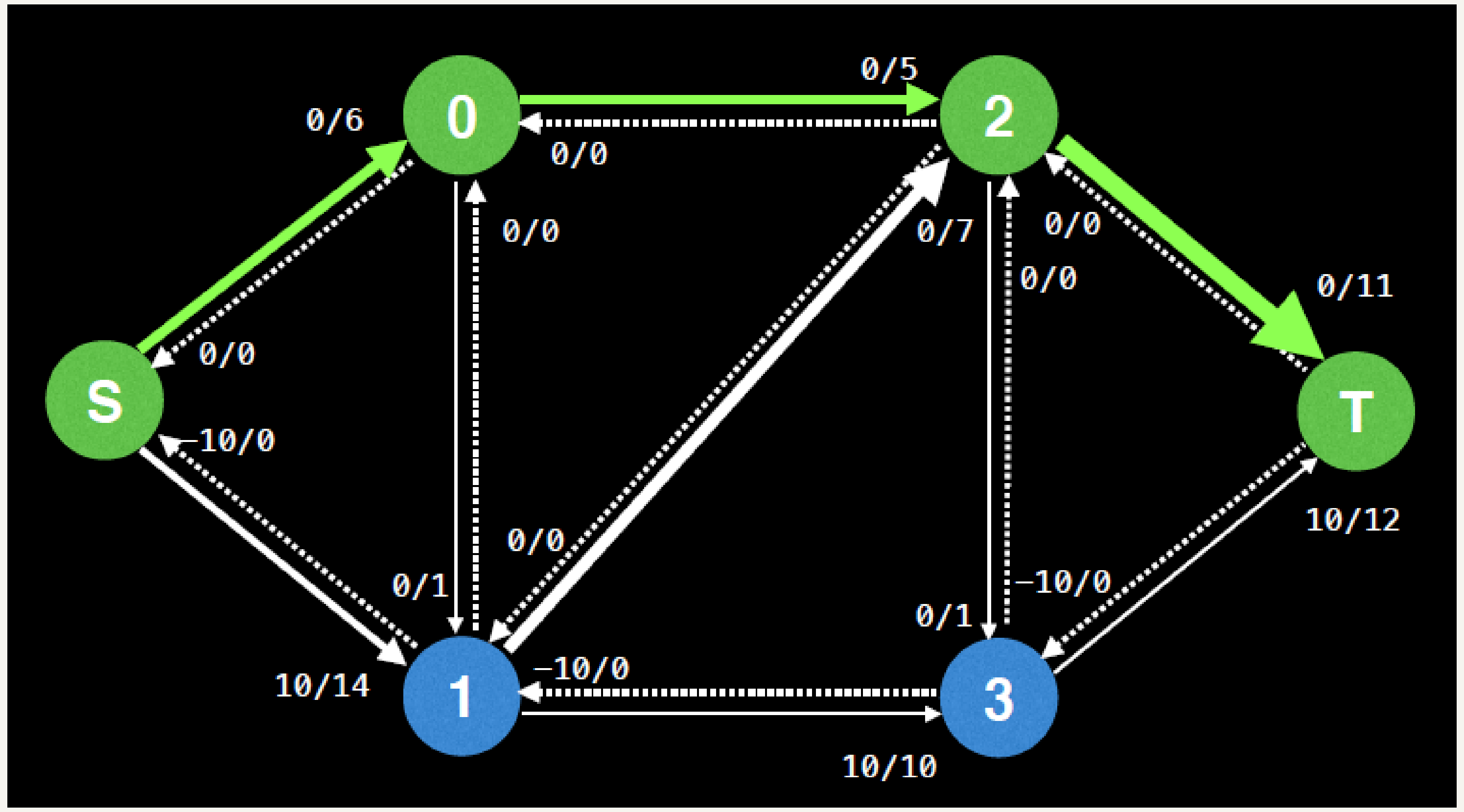


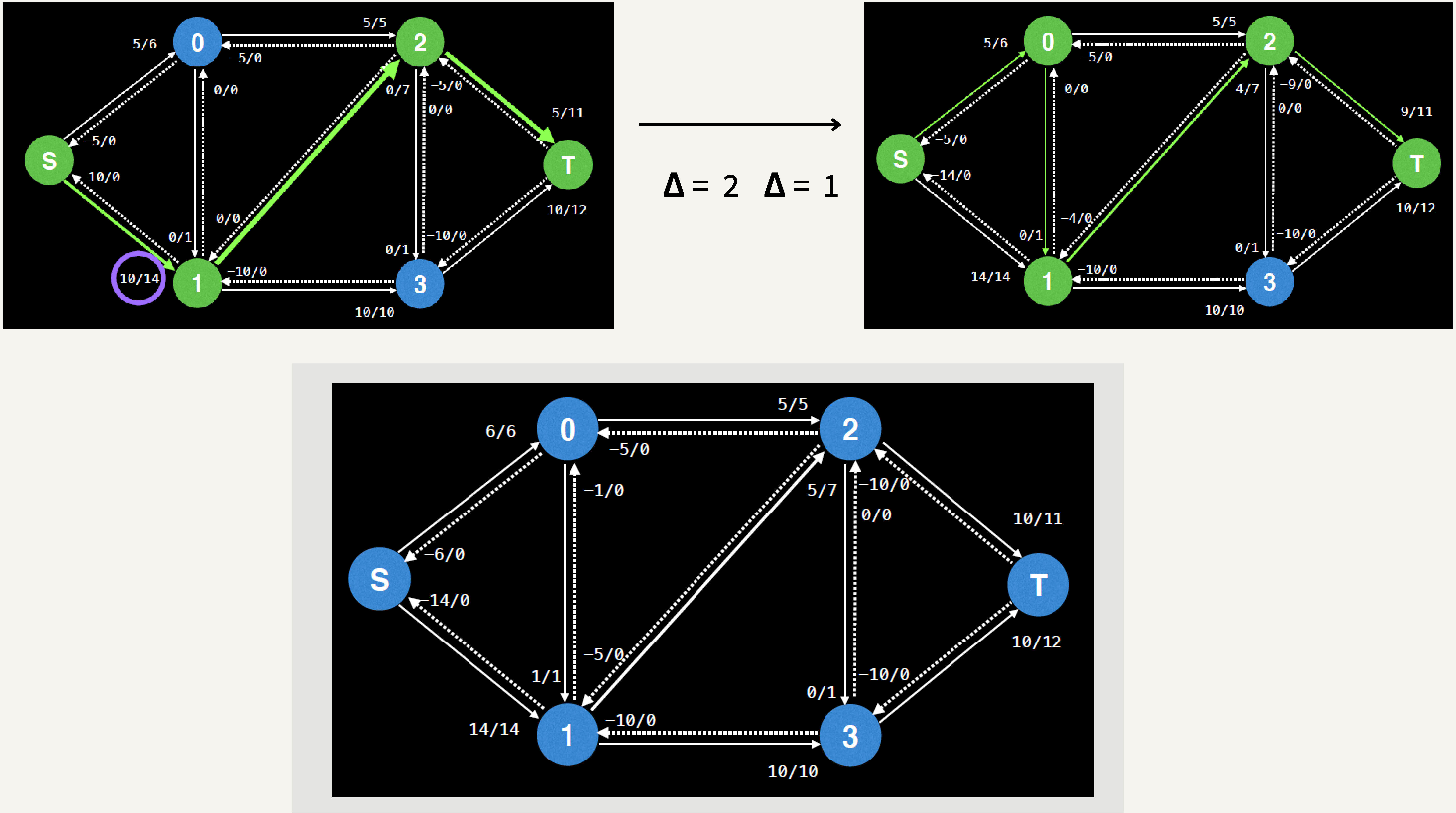
U = 14

Δ = 8



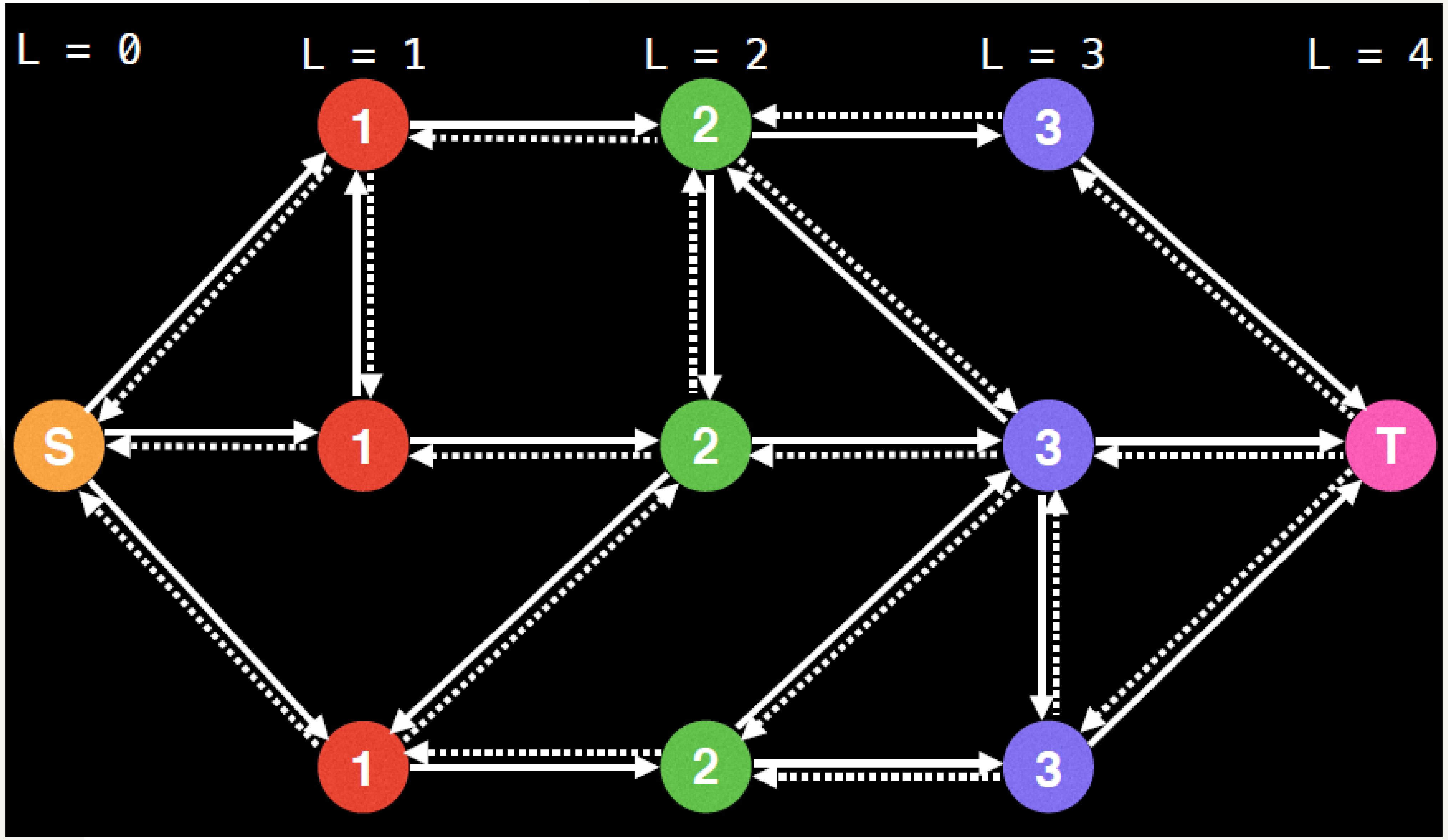
Δ = 4



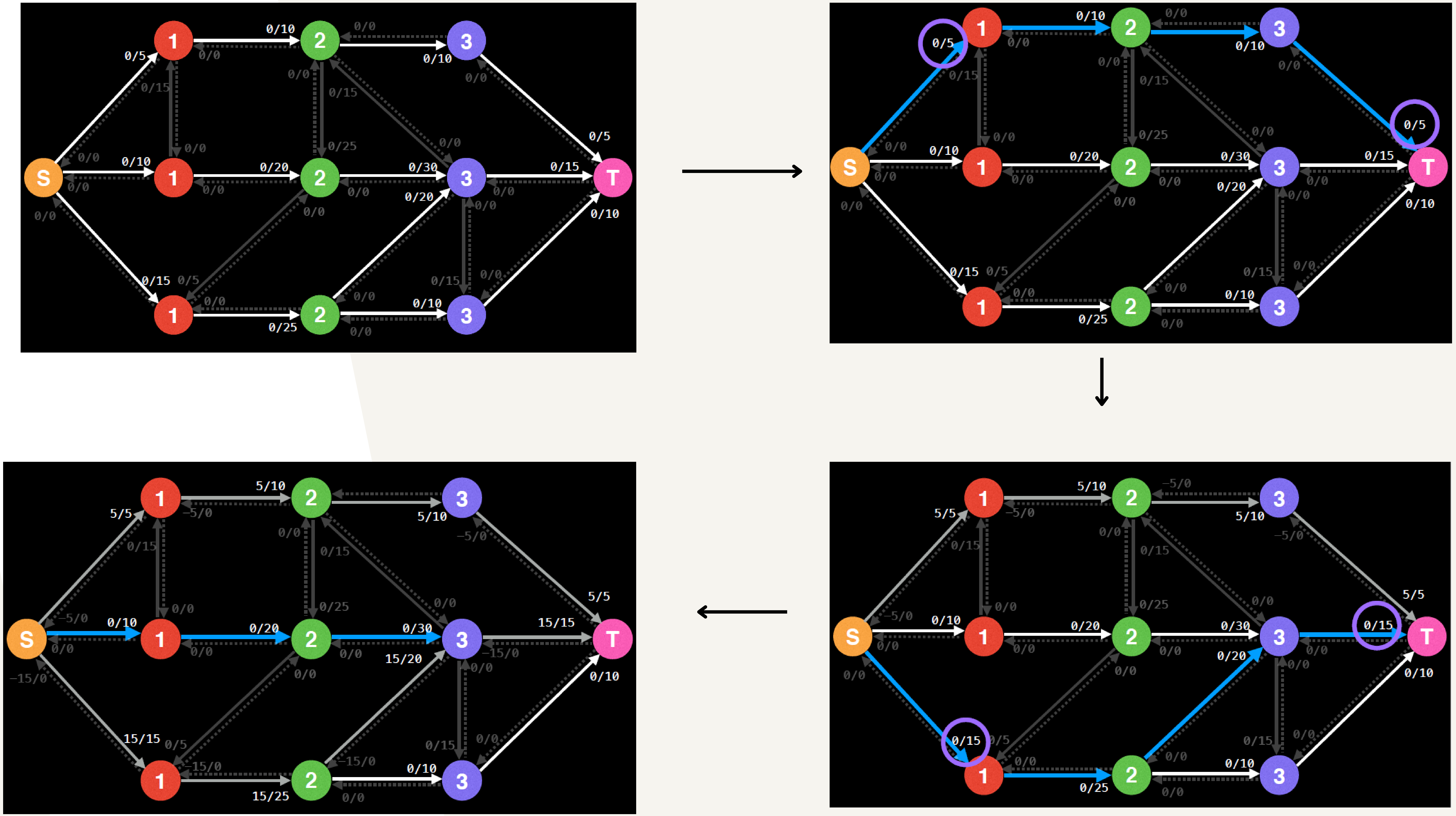


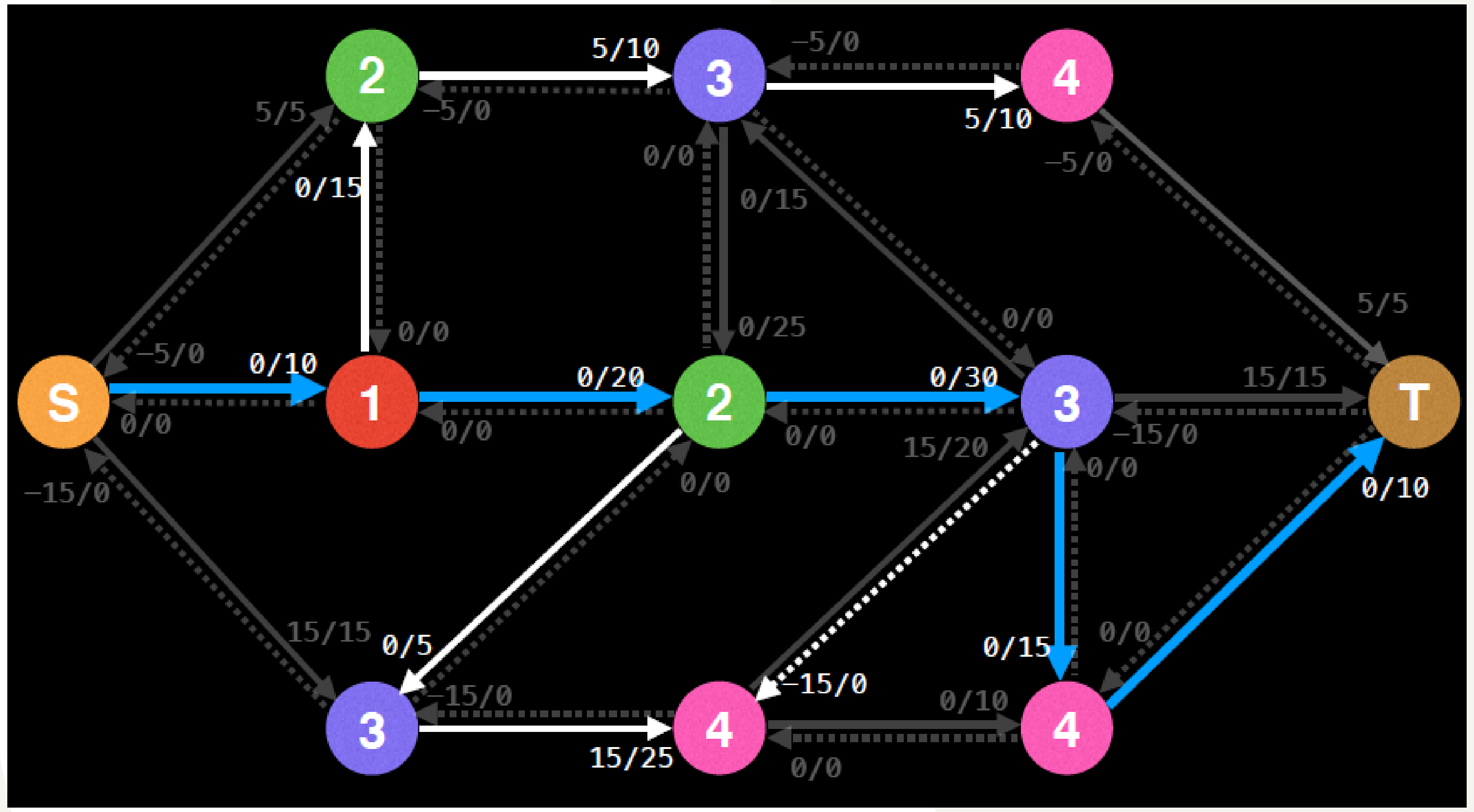
S
0
1
2
3
T
5/6
5/5
−5/0
0/0
0/7
−5/0
0/0
5/11
−5/0
−10/0
0/0
0/1
10/14
−10/0
0/1
−10/0
10/12
10/10
Δ = 2 Δ = 1
5/6
5/5
−5/0
0/0
4/7
−9/0
0/0
9/11
−5/0
−14/0
−4/0
0/1
14/14
−10/0
0/1
−10/0
10/12
10/10
6/6
5/5
−5/0
−1/0
5/7
−10/0
0/0
10/11
−6/0
−14/0
−5/0
1/1
14/14
−10/0
0/1
−10/0
10/12
10/10

Dinic's Algorithm

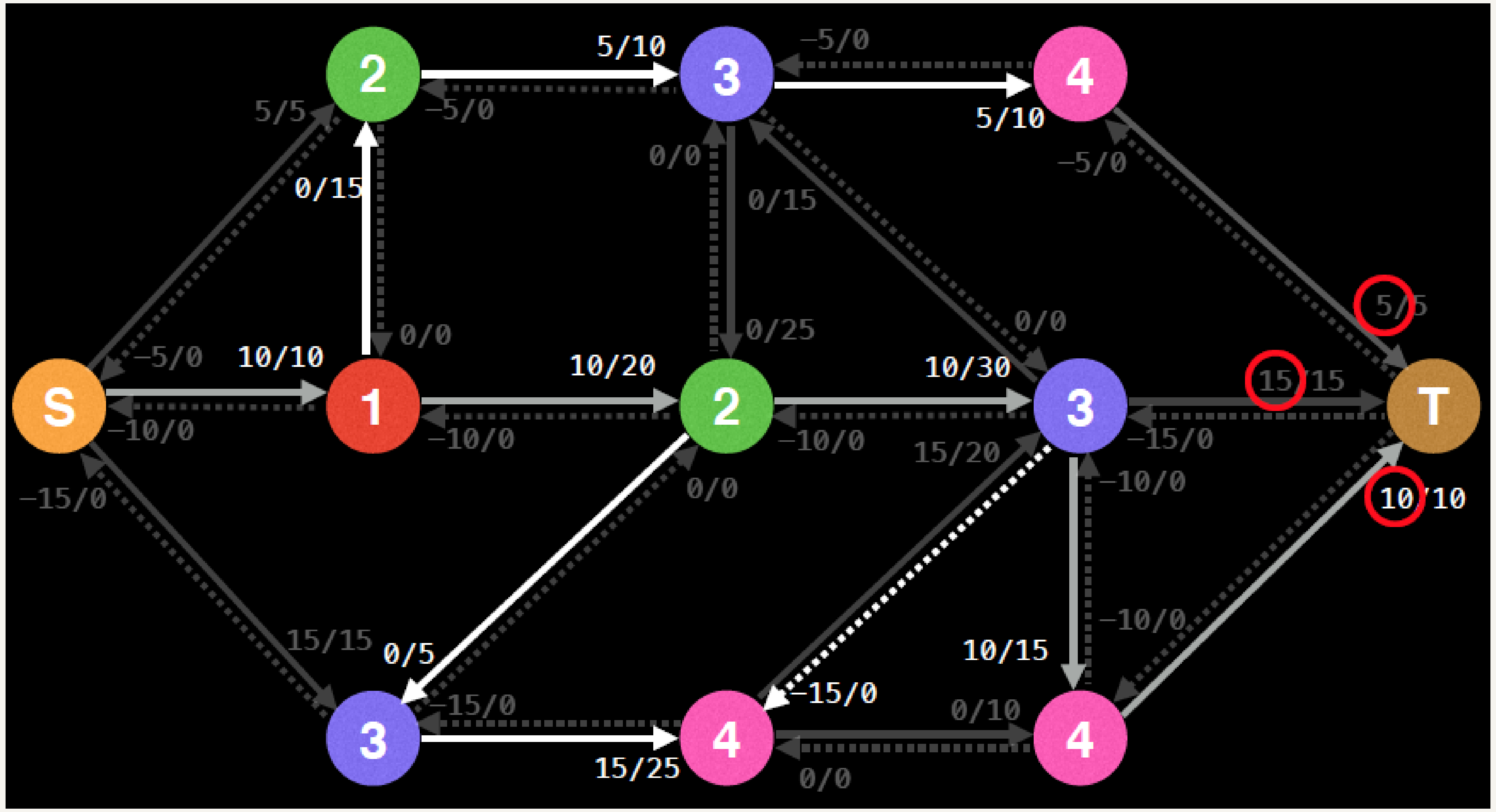


Why take a detour? The main idea behind this algorithm is to guide augmenting paths from source to sink using a level graph and thus reducing the runtime


S
1
2
3
T
0/5
0/10
0/0
0/15
0/25
0/20
0/30
5/5
5/10
-5/0
15/15
15/20
15/25
-15/0



S
1
2
3
4
T
5/5
5/10
-5/0
5/10
-5/0
0/15
0/0
0/15
0/25
0/0
5/5
-5/0
0/10
0/20
0/30
15/15
0/0
0/0
0/0
-15/0
15/20
0/0
0/10
-15/0
15/15
0/5
-15/0
-15/0
0/15
0/0
0/10
15/25
0/0



S
1
2
3
4
T
5/5
5/10
-5/0
5/10
-5/0
0/15
0/0
0/15
0/25
0/0
5/5
-5/0
10/10
10/20
10/30
15/15
-10/0
-10/0
-10/0
-15/0
15/20
0/0
-10/0
10/10
-15/0
15/15
0/5
-15/0
-15/0
10/15
-10/0
0/10
15/25
0/0

Conclusion

Algorithm	Time Complexity
Ford Fulkerson (with DFS)	$O(EF)$, where F is the maximum flow
Edmonds-Karp	$O(VE^2)$
Dinic's algorithm	$O(V^2E)$
Capacity Scaling	$O(VE \log U)$, where U is the maximum capacity of the network

Applications

Transportation Networks - Optimizing the flow of traffic, passengers or goods through the network minimizing congestion and maximizing efficiency.

Disaster Mangement - It can be used to determine evacuation routes, resource allocation and logistics planning during emergencies

Image Segmentation - Treating image as a graph, the algorithm finds the optimal boundary between different regions based on flow of pixels

Water Management- Determining the optimal distribution of water resources in irrigation networks, hydroelectric power generation, or water supply networks

Thank You