Design and Analysis of Algorithms

- ***Course Code***: BCSE204L
- ***Course Type***: Theory (ETH)
- ***Slot***: A1+TA1 & & A2+TA2
- ***Class ID***: VL2023240500901
  VL2023240500902

A1+TA1

Day	Start	End
Monday	08:00	08:50
Wednesday	09:00	09:50
Friday	10:00	10:50

A2+TA2

Day	Start	End
Monday	14:00	14:50
Wednesday	15:00	15:50
Friday	16:00	16:50

Syllabus- Module 4

Module:4	**Graph Algorithms**	**6 hours**

All pair shortest path: Bellman Ford Algorithm, Floyd-Warshall Algorithm
Network Flows: Flow Networks, Maximum Flows: Ford-Fulkerson, Edmond-Karp, Push Re-label Algorithm – Application of Max Flow to maximum matching problem

What is Network Flows, Flow Networks and Maximum Flows

Network flows, flow networks, and maximum flows are fundamental concepts in graph theory and optimization, particularly in the context of transportation, communication, and resource allocation problems.

Network Flows:

Network flows refer to the movement of resources, such as goods, information, or energy, through a network.

This network can be represented as a directed graph, where Nodes represent locations or entities, Edges represent connections or pathways between them.

The flow along each edge represents the quantity of resources being transferred from one node to another.

Network flows are used to model various real-world scenarios, including transportation networks, communication networks, and supply chain logistics.

What is Network Flows, Flow Networks and Maximum Flows

Network flows, flow networks, and maximum flows are fundamental concepts in graph theory and optimization, particularly in the context of transportation, communication, and resource allocation problems.

Flow Networks:

Flow networks, also known as flow graphs or transportation networks, are directed graphs used to model and analyze the flow of resources through a network.

A flow network consists of a set of nodes (vertices) and directed edges connecting these nodes. Each edge is associated with a capacity, which represents the maximum amount of flow that can pass through it. Additionally, flow networks contain special nodes known as the source and the sink.

The source is the node from which the flow originates, while the sink is the node that absorbs the flow. Flow networks are characterized by their ability to represent and solve flow optimization problems efficiently.

Some real-life problems like the flow of liquids through pipes, the current through wires and delivery of goods can be modelled using flow networks.

Flow Network Definition:

- A directed graph $G=(V,E)$ where V represents vertices and E represents edges.
- Each edge $(u,v)\in E$ is associated with a nonnegative weight capacity $c(u,v)\geq 0$.
- There are two distinguished vertices: the **source** s and the **sink** t.
- Every vertex $v\in V$ has a path from s to t.

Flow Networks

formal definition of **Flow Network Definition**:

- A directed graph $G=(V,E)$ where V represents vertices and E represents edges.
- Each edge $(u,v)\in E$ is associated with a nonnegative weight capacity $c(u,v)\geq 0$.
- There are two distinguished vertices: the source s and the sink t.
- Every vertex $v\in V$ has a path from s to t.

- A flow $f:V\times V\rightarrow R$ is a real-valued function satisfying certain properties.
- **Capacity Constraint**: $f(u,v)\leq c(u,v)$ for all $u,v\in V$.
- **Skew Symmetry**: $f(u,v)=-f(v,u)$ for all $u,v\in V$.
- **Flow Conservation**: Total net flow out of any vertex other than s and t is zero.
    - For all $u\in V-\{s,t\}$, the net flow out of u is zero.

What is Network Flows, Flow Networks and Maximum Flows

Network flows, flow networks, and maximum flows are fundamental concepts in graph theory and optimization, particularly in the context of transportation, communication, and resource allocation problems.

In the **maximum-flow problem**, we are given a flow network G with source s and sink t, and we wish to find a flow of maximum value from s to t.

Maximum Flows: Maximum flows refer to the maximum amount of flow that can be sent from the source to the sink in a flow network. The maximum flow problem is a classic optimization problem that aims to determine the maximum flow value achievable in a given flow network.

Mathematically, it involves finding a flow distribution that satisfies the capacity constraints of the network while maximizing the total flow from the source to the sink. Maximum flow algorithms are used to efficiently solve this problem and find the optimal flow distribution. Maximum flows have numerous applications, including network routing, capacity planning, and resource allocation in various industries.

- Capacity Constraint ensures that the flow through each edge does not exceed its capacity.
- Skew Symmetry indicates that the flow from u to v is the negative of the flow from v to u.
- Flow Conservation ensures that the total net flow out of any vertex (except s and t) is zero, meaning the flow into a vertex equals the flow out of it.

Maximum flow algorithms: Ford-Fulkerson

- The Ford-Fulkerson algorithm, developed by L.R. Ford and Dr. R. Fulkerson in 1956, is a fundamental method for finding the maximum flow in a network.

Before delving into the algorithm, it's essential to define several key concepts for better comprehension:

1. **Residual Capacity**: The residual capacity of a directed edge is the difference between the edge's capacity and the current flow through it.

(capacity of the edge - current flow through the edge)

If there's a flow along a directed edge from node u to v, the reversed edge has a capacity of 0, and we denote it as f(v,u) = −f(u,v).

2. **Residual Graph**: The residual graph is derived from the original graph with the distinction that it employs residual (Remaining) capacities as edge capacities.

3. **Augmenting Path**: An augmenting path is a simple path from the source (S) to the sink (T) in a residual graph, traversing edges with a capacity of 0.

4. **Minimal Cut**-Bottneck capacity, which decides maximum flow from source to sink through augment path.

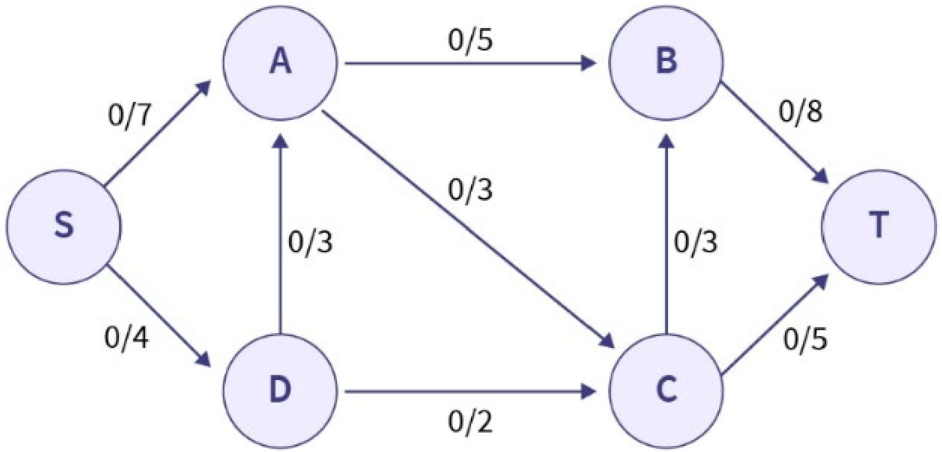
Maximum flow algorithms: **Ford-Fulkerson**

1. Initialize the flow of each edge e to 0.
2. Search for an augmenting path between the source (s) and the sink (t) in the residual graph.
3. If such a path exists, increase the flow along those edges.
4. Repeat steps 2-3 until no more augmenting paths exist in the residual graph. At this point, the maximum flow is achieved.

Maximum flow algorithms: Ford-Fulkerson (Example)

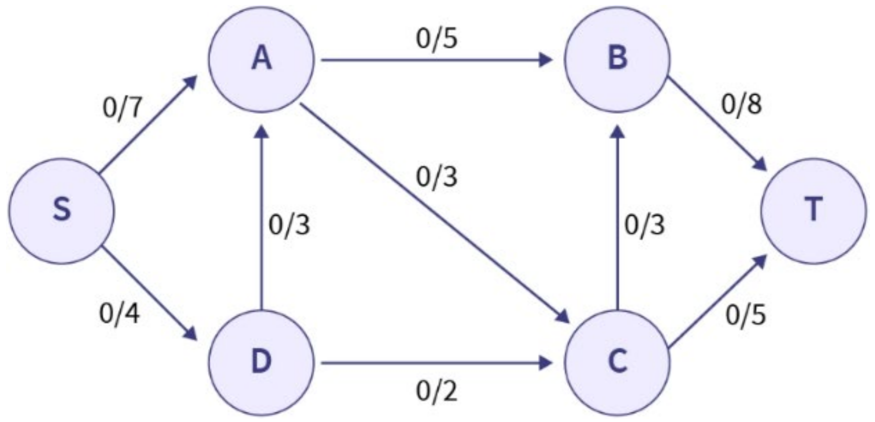
	S	A	B	C	D	T
S	0	7	inf	inf	4	inf
A	inf	0	5	3	inf	inf
B	inf	inf	0	inf	inf	8
C	inf	inf	3	0	inf	5
D	inf	3	inf	2	0	inf
T	inf	inf	inf	inf	inf	0

Start with all edge flows set to 0.

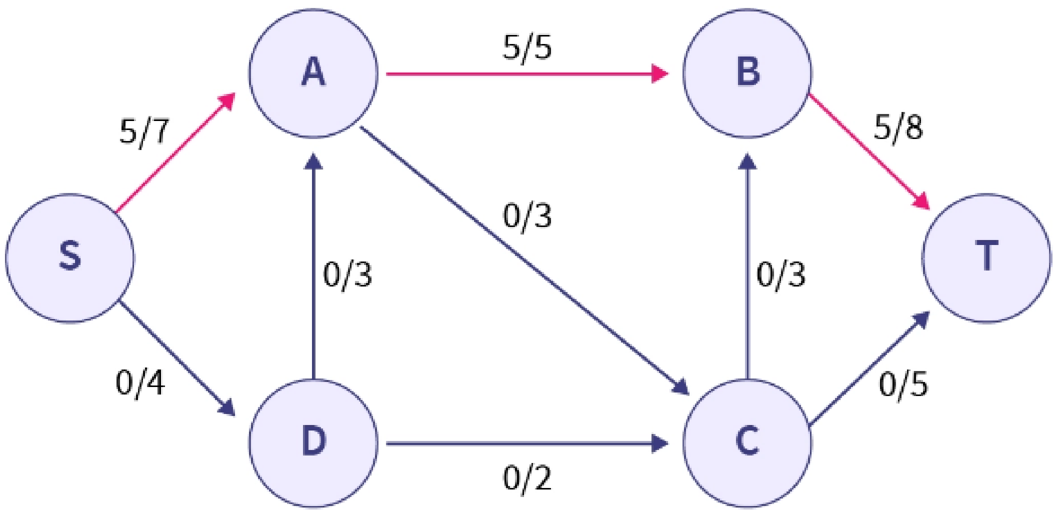


Maximum flow algorithms: Ford-Fulkerson (Example)

Start with all edge flows set to 0.



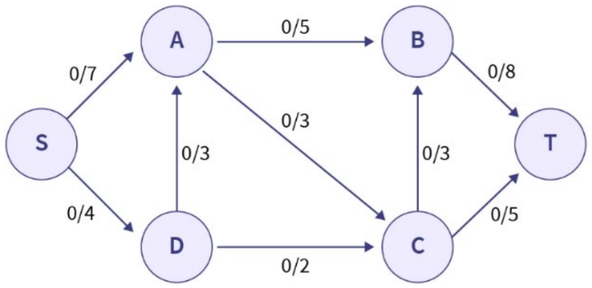
Search for an augmenting path. One such path is **s→A→B→t** with residual capacities of 7, 5, and 8. The minimum is 5, so increase the flow along the path by 5.



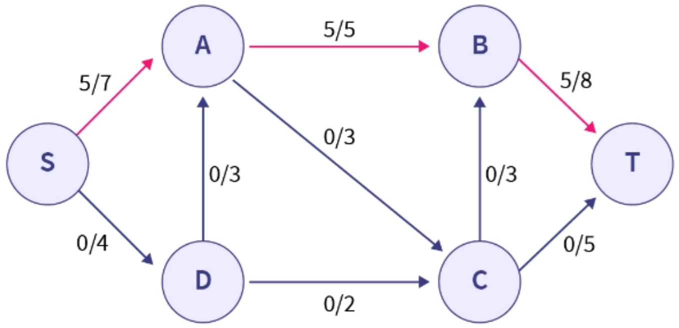
Repeat until no more augmenting paths are found.

Maximum flow algorithms: Ford-Fulkerson (Example)

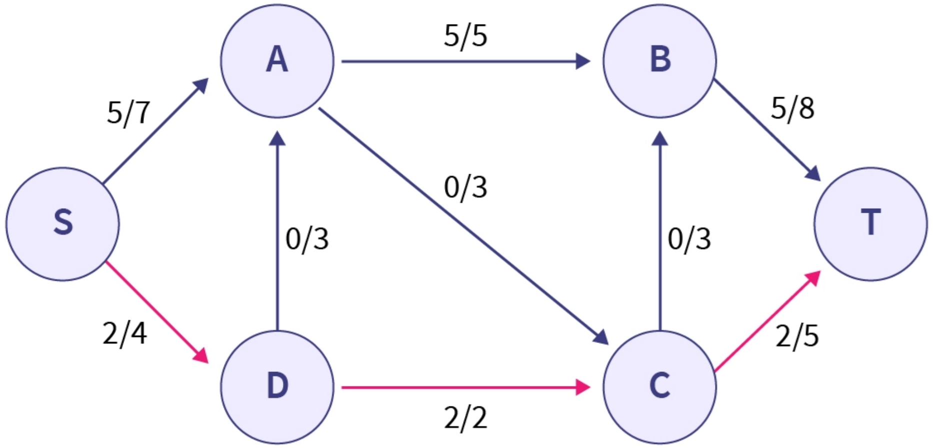
Start with all edge flows set to 0.



Search for an augmenting path. One such path is **s→A→B→t** with residual capacities of 7, 5, and 8. The minimum is 5, so increase the flow along the path by 5.

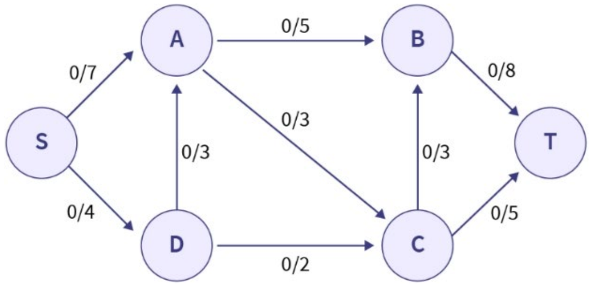


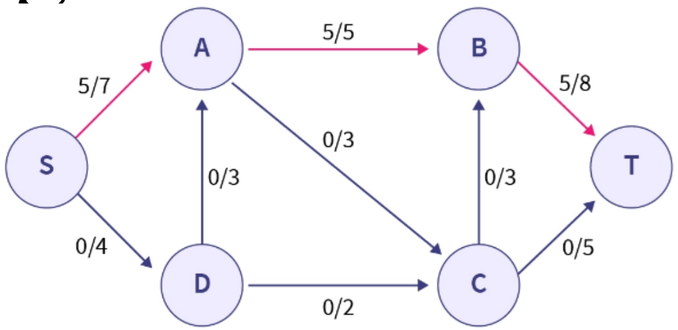
Look for other possible augmenting paths. For example, s→D→C→t with residual capacities of 4, 2, and 5, with 2 being the minimum. Increase the flow by 2 along this path.

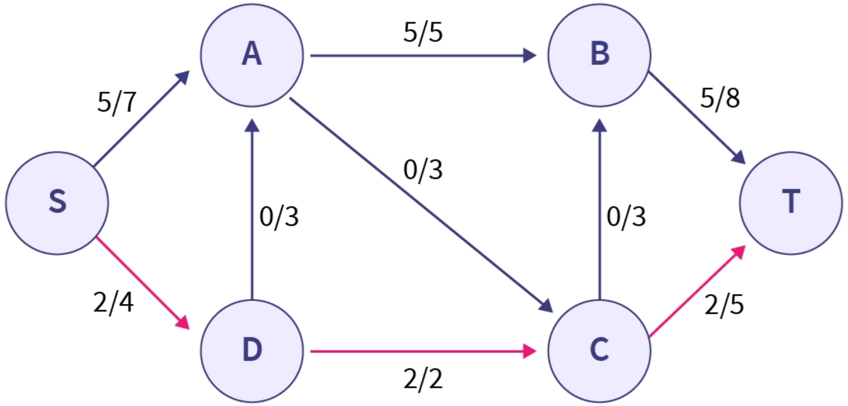


Repeat until no more augmenting paths are found.

Maximum flow algorithms: Ford-Fulkerson (Example)

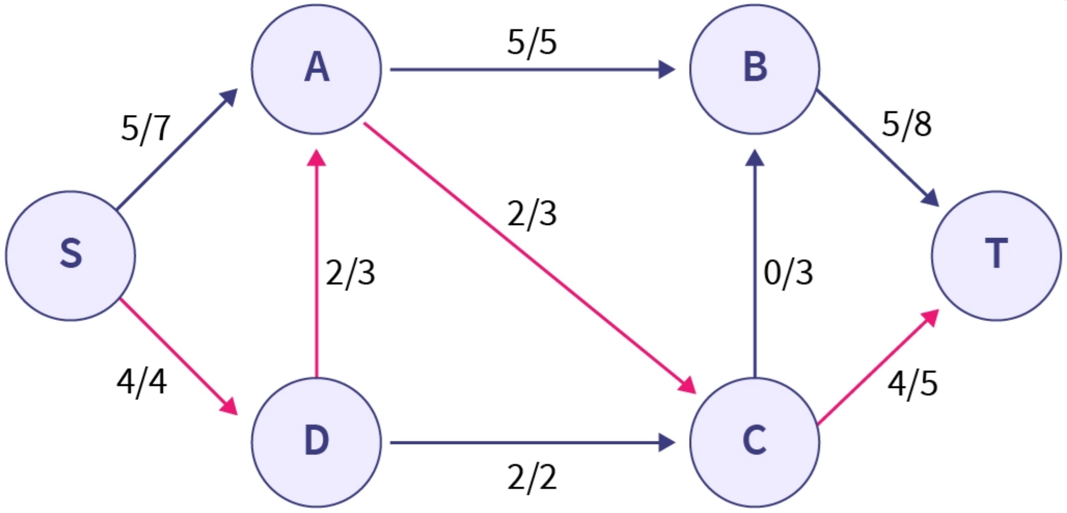






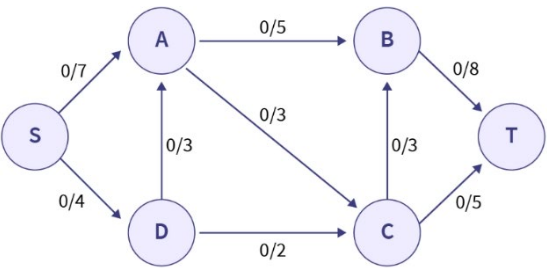
Continue searching for augmenting paths, ensuring that previously saturated edges (such as A→B and D→C are flow has reached their maximum capacity) are not included.

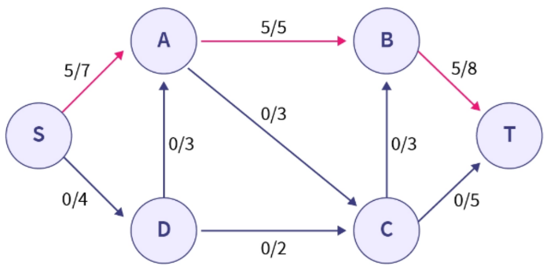
For instance, **s→D→A→C→t** has residual capacities of 2, 3, 3, and 3, with 2 being the minimum.
Increase the flow by 2 along this path.

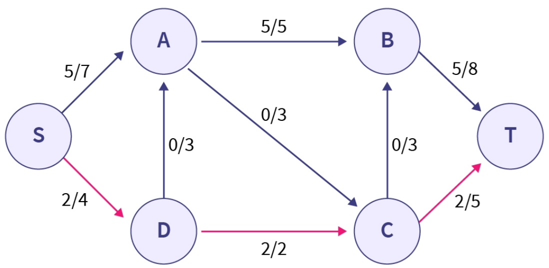


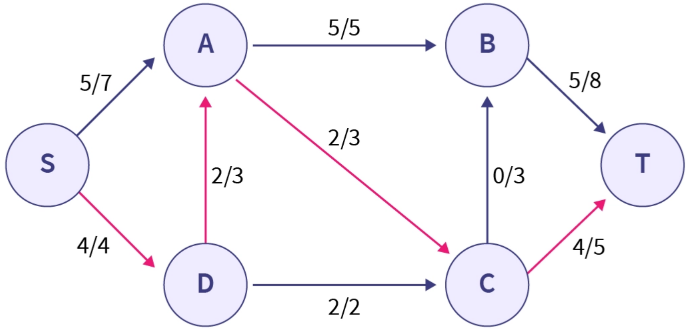
Repeat until no more augmenting paths are found.

Maximum flow algorithms: Ford-Fulkerson (Example)



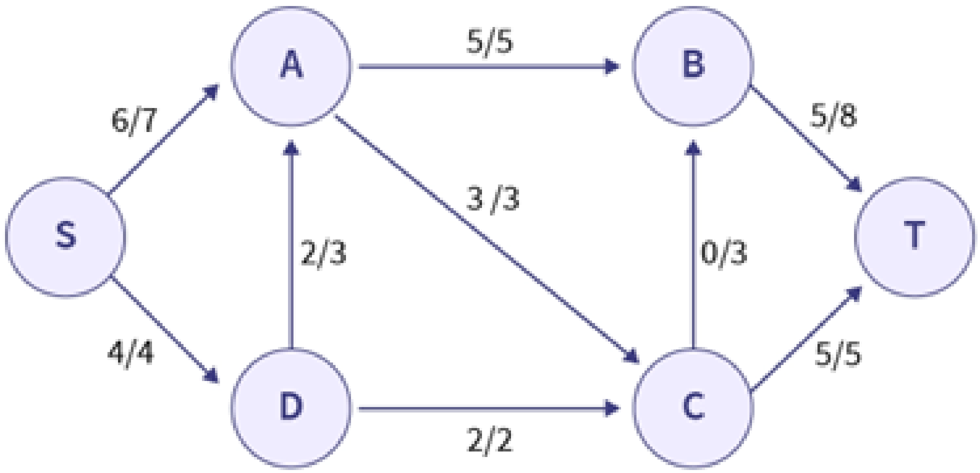






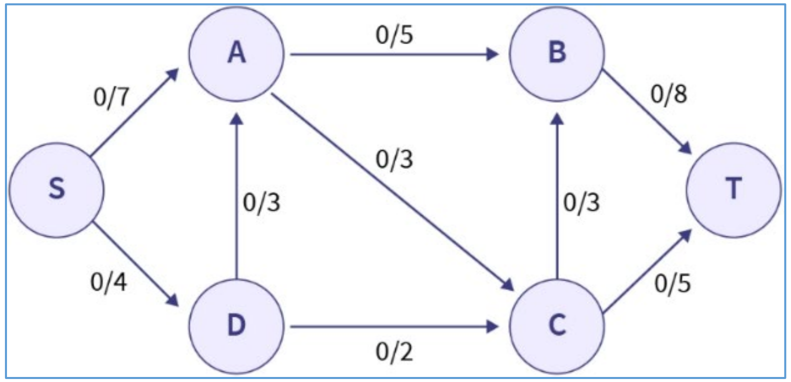
Continue searching for augmenting paths, ensuring that previously saturated edges (such as A→B and D→C are flow has reached their maximum capacity) are not included.

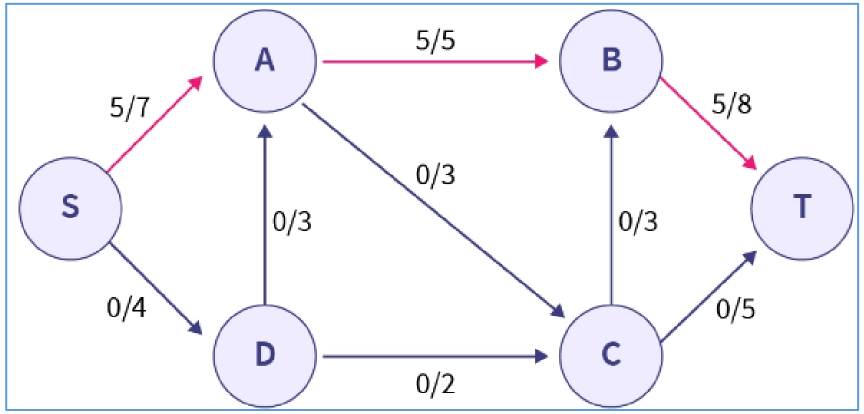
For instance, **s→A→C→t** has residual capacities of 2, 1, and 1, with 1 being the minimum. Increase the flow by 2 along this path.

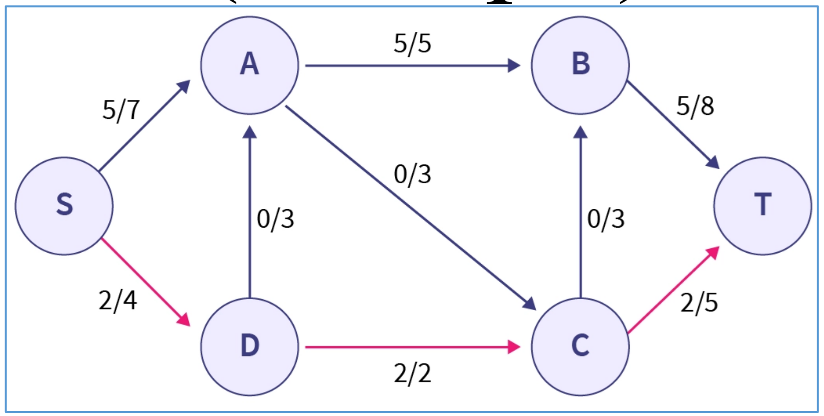


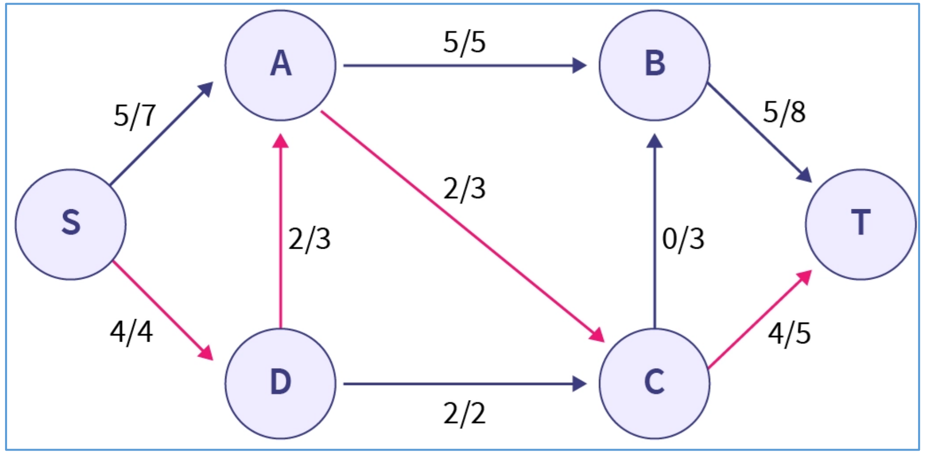
Now there are no more augmenting paths possible in the network

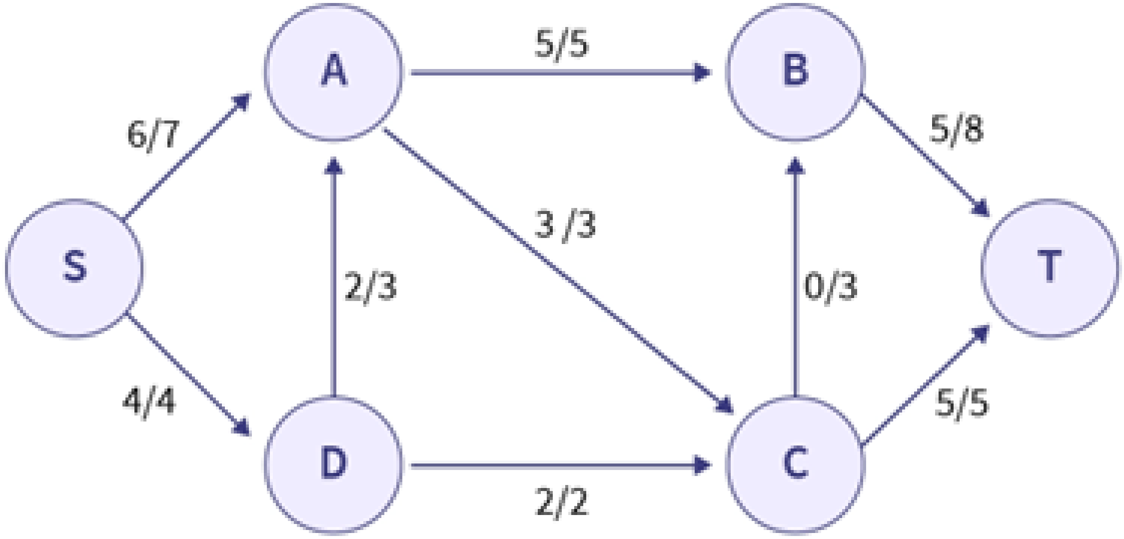
Maximum flow algorithms: Ford-Fulkerson (Example)







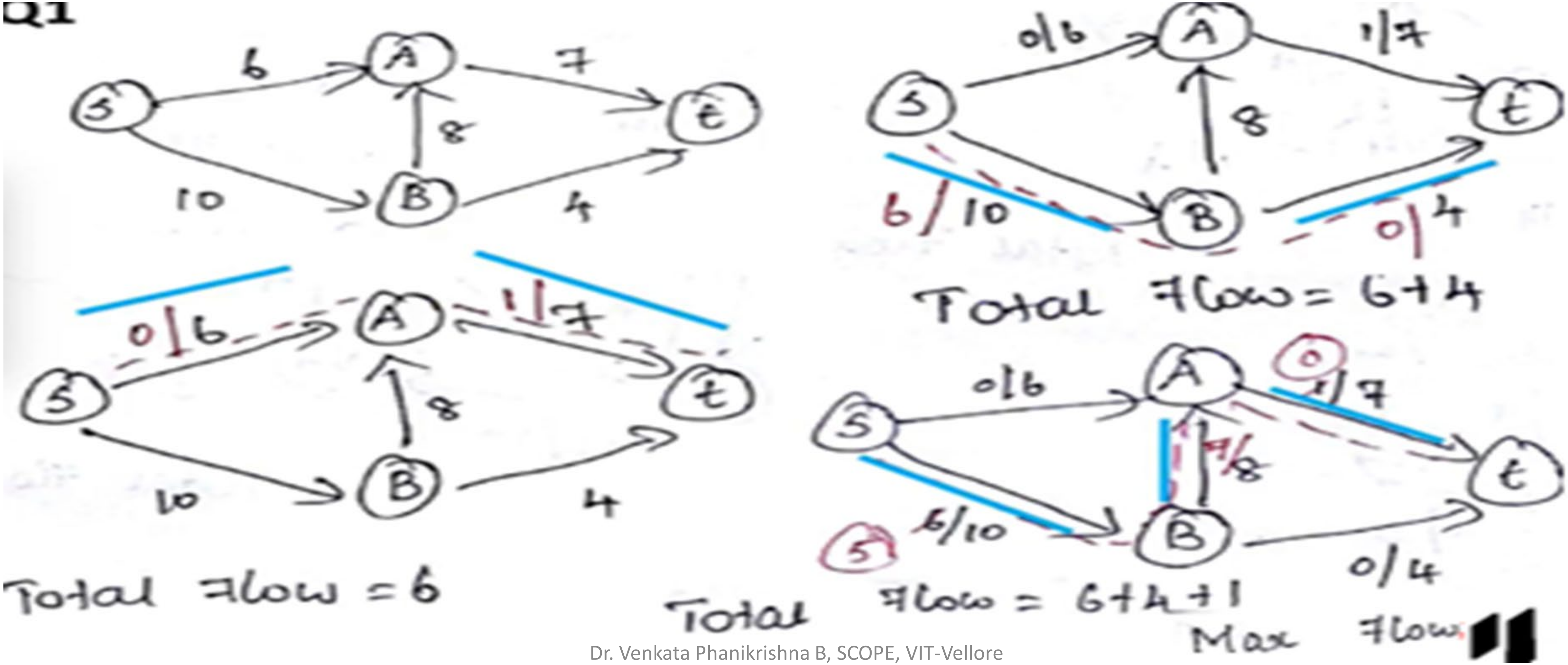


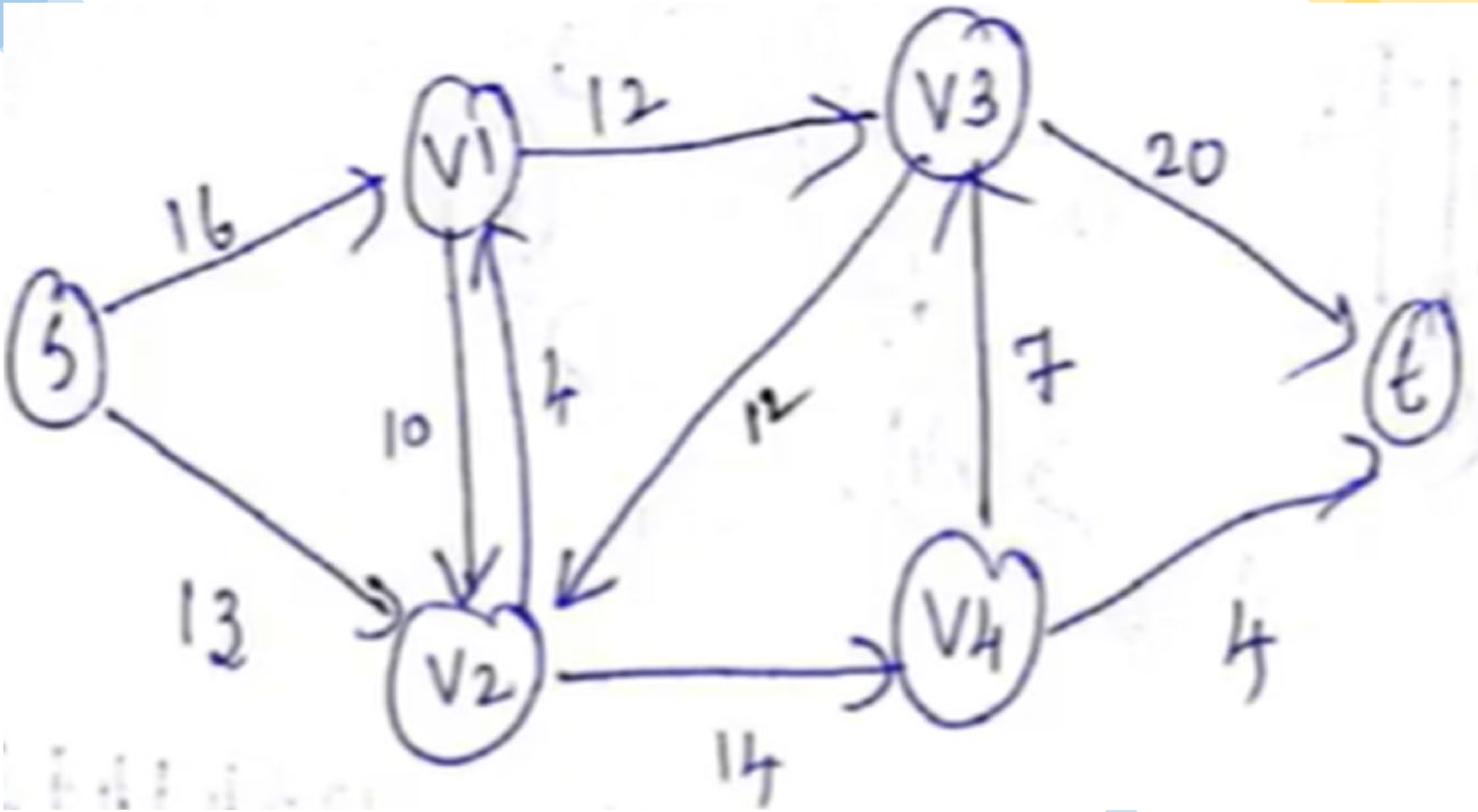


The maximum flow is determined by the total flow going out of the source or coming into the sink. In this case, it is 6+4=5+5=10.

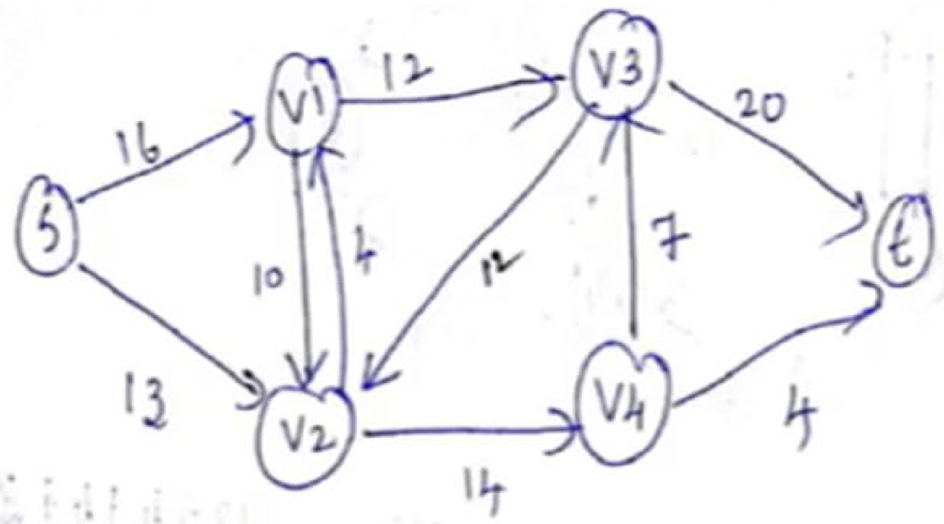
Max flow =11

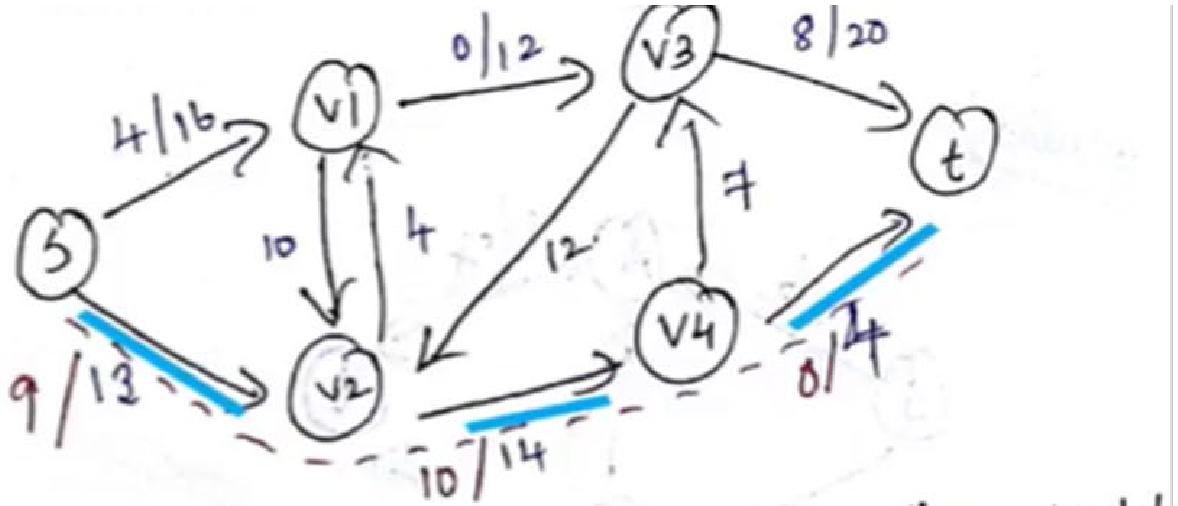
Q1





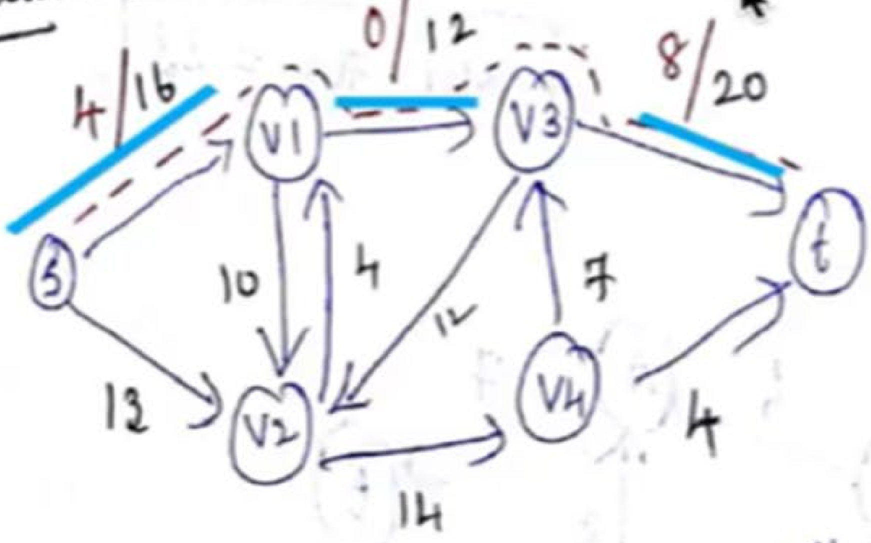
s
V1
V2
V3
V4
t
16
13
10
4
12
12
14
7
20
4



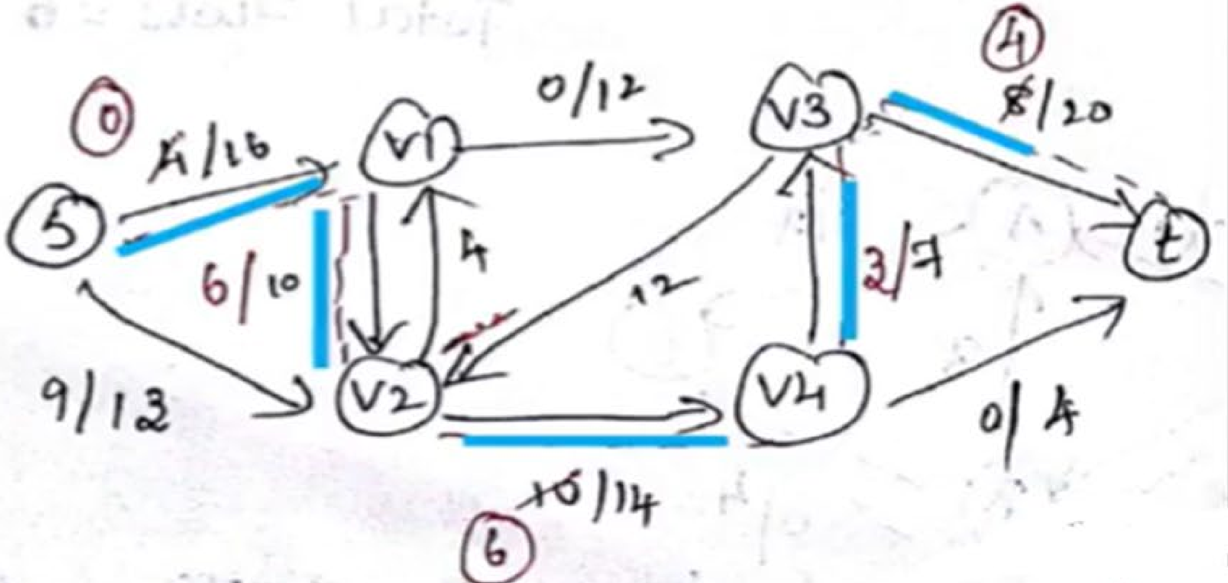


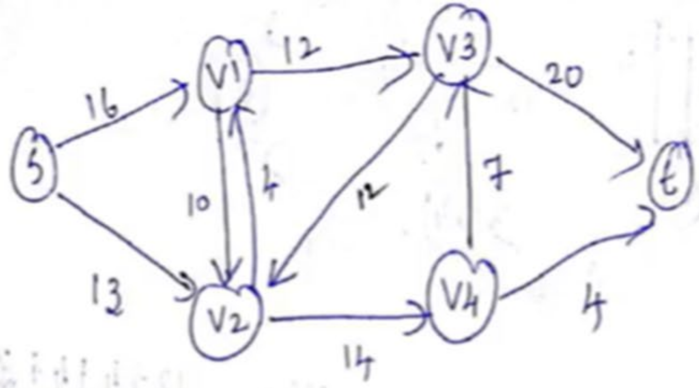
Total Flow = 12+4

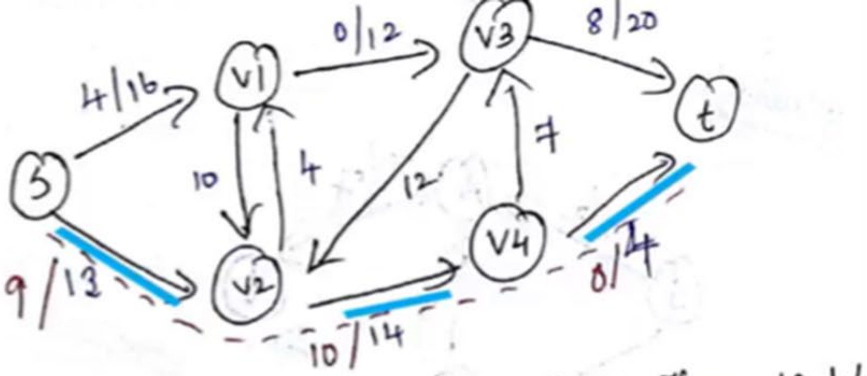
Solution:



Total Flow = 12

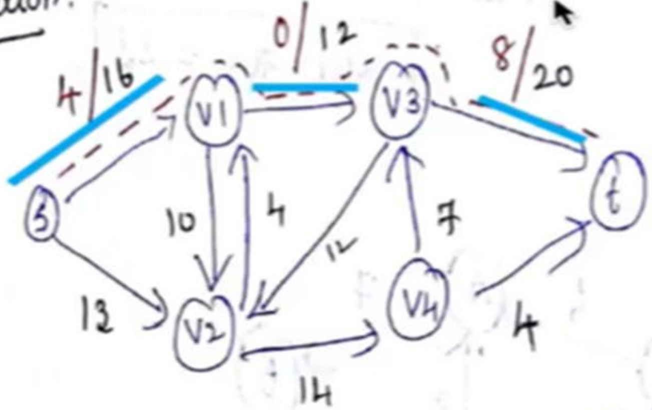




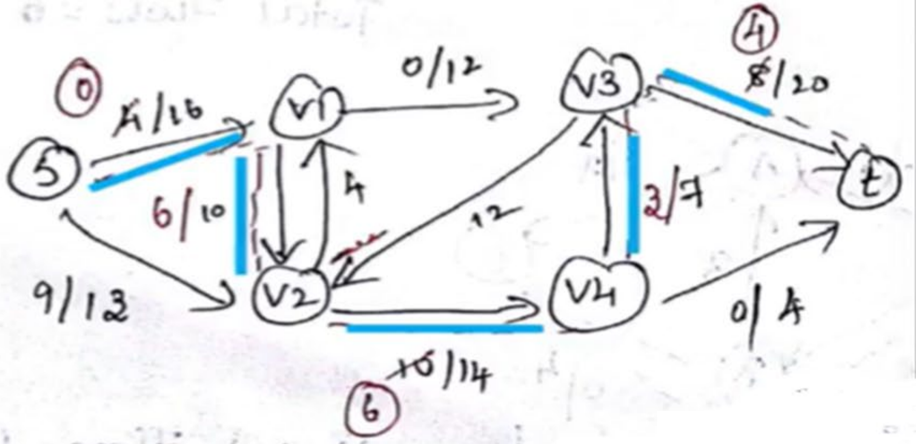


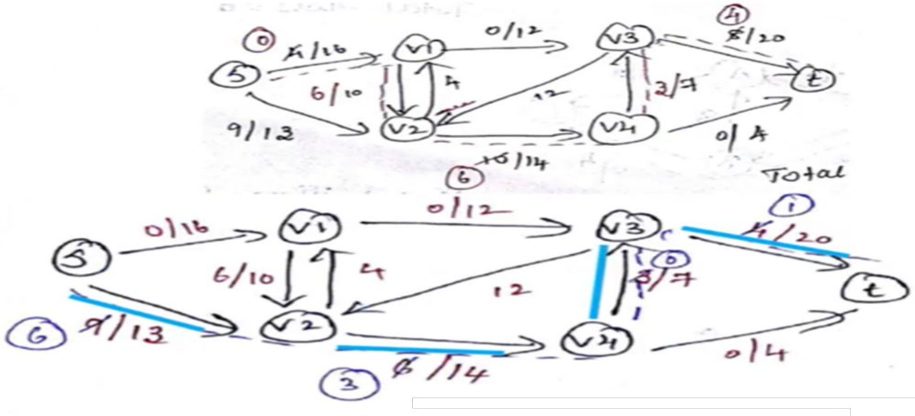
Total Flow = 12 + 4

Solution:



Total Flow = 12





Max Total Flow = 12 + 4 + 4 + 3

$\boxed{\text{Max Flow} = 23}$

Algorithm:

FordFulkerson(Graph G, Node s, Node t):

1. Initialize flow of all edges e to 0.
2. while(there is augmenting path(P) from s to t
3. in the residual graph):

    Find augmenting path between s and t.

    Update the residual graph.

    Increase the flow.

4. return total flow

Complexity

The complexity of Ford-Fulkerson is : **O(E F)**

Where F is the maximal flow of the network and E is number of edges.

Applications

- In traffic movements, to find how much traffic can move from a City A to City B.
- In electrical systems, to find the maximum current that could flow the wires without causing any short circuit.
- In the water/sewage management system, to find maximum capacity of the network.


PresenterMedia


Any
Question


Thank You!


FOR YOUR
ATTENTION