Graph-Cut for Image Reconstruction and Segmentation

Dr. V. Masilamani
Dept. of Computer Science & Engg.
IIITDM Kancheepuram
Chennai-600127, Tamilnadu.

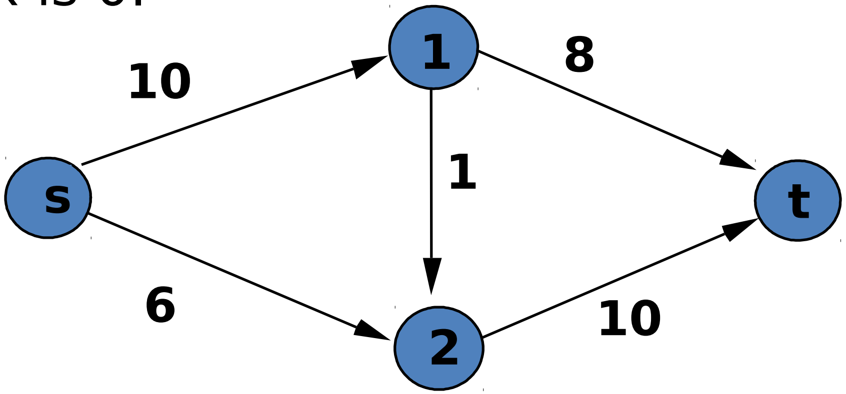
Outline of the Talk

- Flow Network
- Ford-Fulkerson Algorithm to Compute Max-Flow
- Max-Flow Min-Cut Theorem
- Graph Vs. Image
- Image Reconstruction using Max-Flow
- Image Segmentation using Min-Cut
- Summary

Flow Network

- **Definition of flow network**

  Flow network is a directed weighted graph G=(V,E,c) such that
  1) Weight(capacity) $c(u,v) \geq 0$.
  2) Two distinguished vertices exist in G namely :
     - Source (denoted by s) : In-degree of this vertex is 0.
     - Sink (denoted by t)   :   Out-degree of this vertex is 0.



Flow

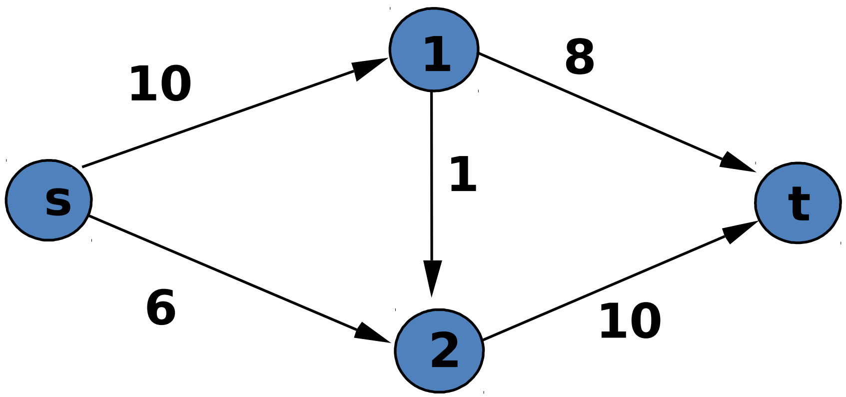
- **Definition of flow**

  Flow in a network is an integer-valued function f defined on the edges of G satisfying

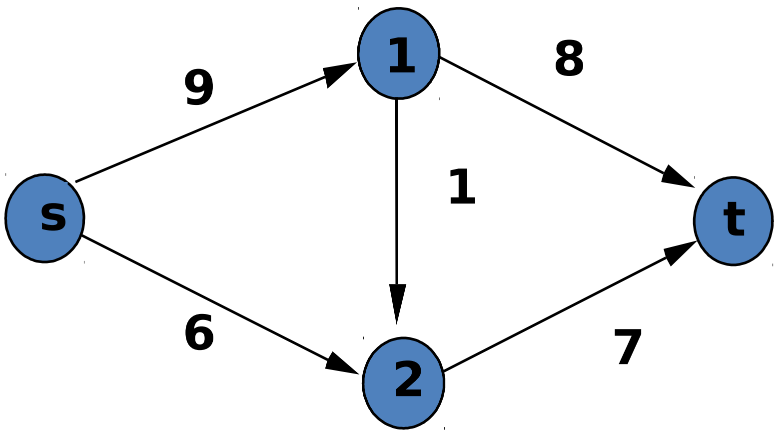
  1) $0 \le f(u,v) \le c(u,v)$, for every edge (u,v) in E.
  2) Capacity Constraint : $\forall\, u,v \in V,\ f(u,v) \le c(u,v)$
  3) Skew Symmetry : $\forall\, u,v \in V,\ f(u,v) = -f(v,u)$
  4) Flow Conservation : For each vertex v, inflow(v)=outflow(v)

  Skew symmetry condition implies that f(u,u)=0.

Flow Network:



Flow:



Max-Flow

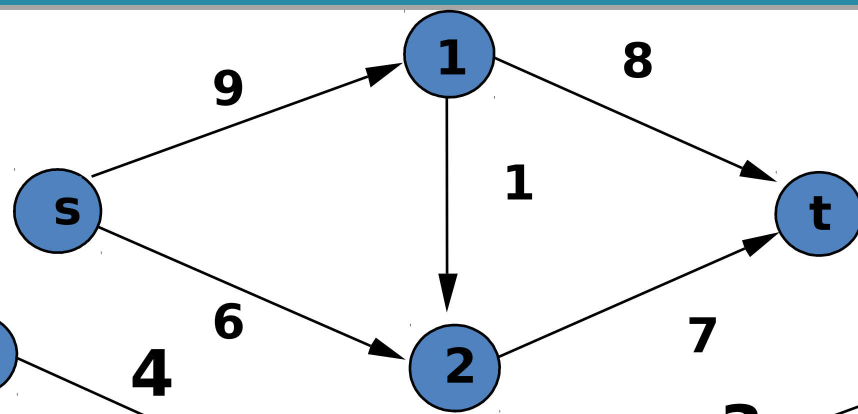
- The value of a flow is given by :

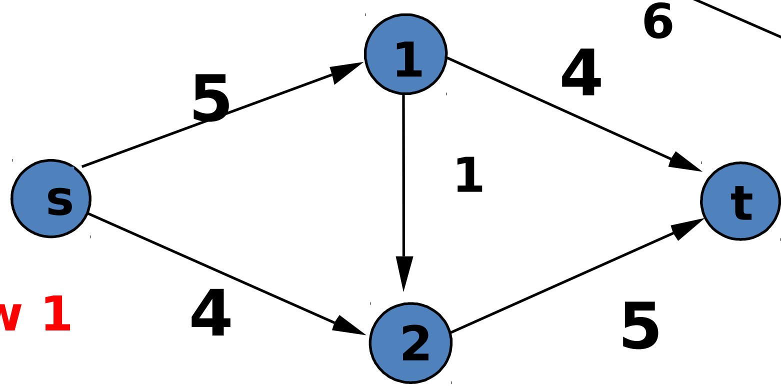
$$|f| = \sum_{v \in V} f(s,v) = \sum_{v \in V} f(v,t)$$

- Definition of Max-flow
  - Given a graph G=(V, E) with capacities on edges, find flow f, such that |f| is maximum.

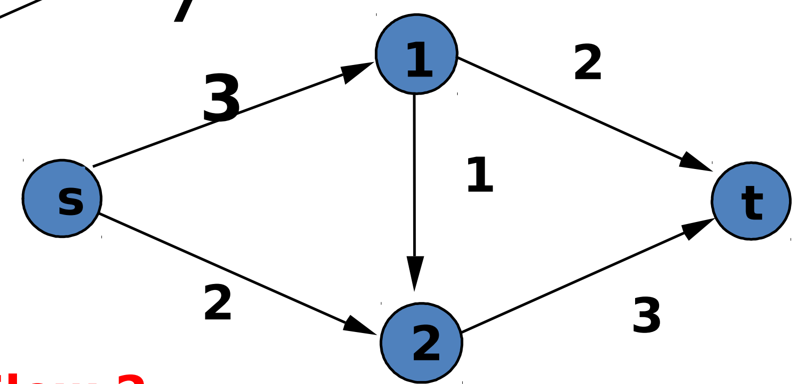
Max-Flow

Network

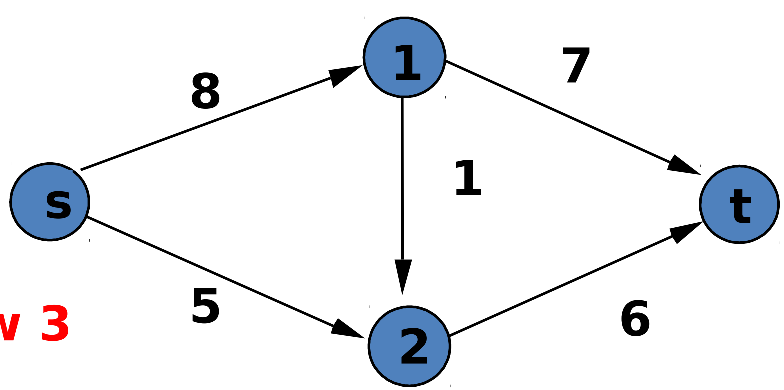




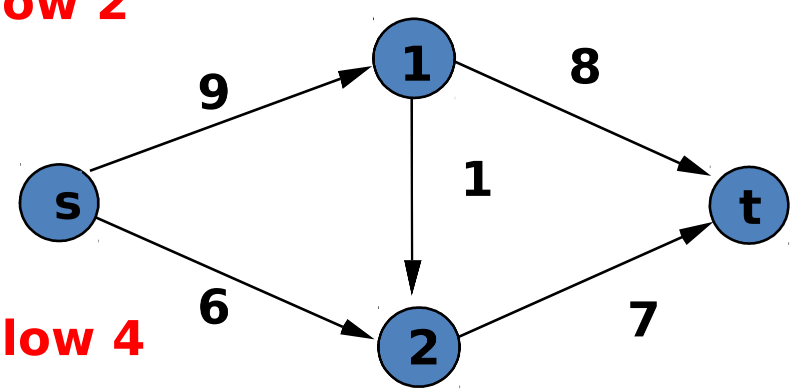
Flow 1



Flow 2

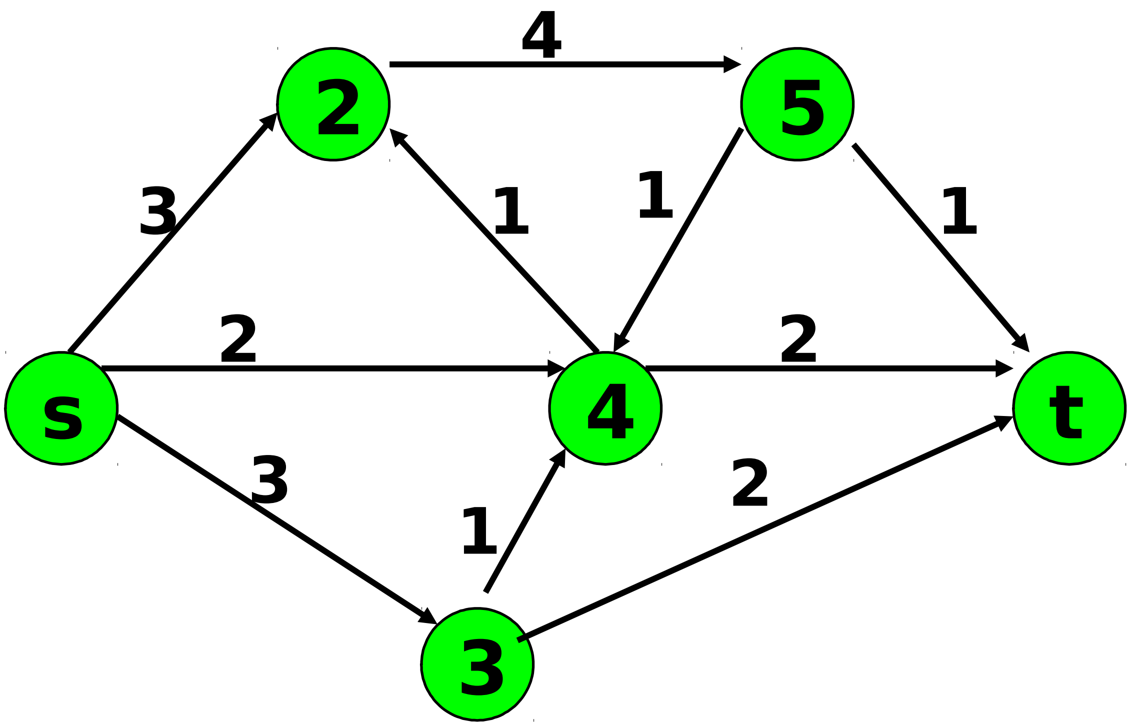


Flow 3



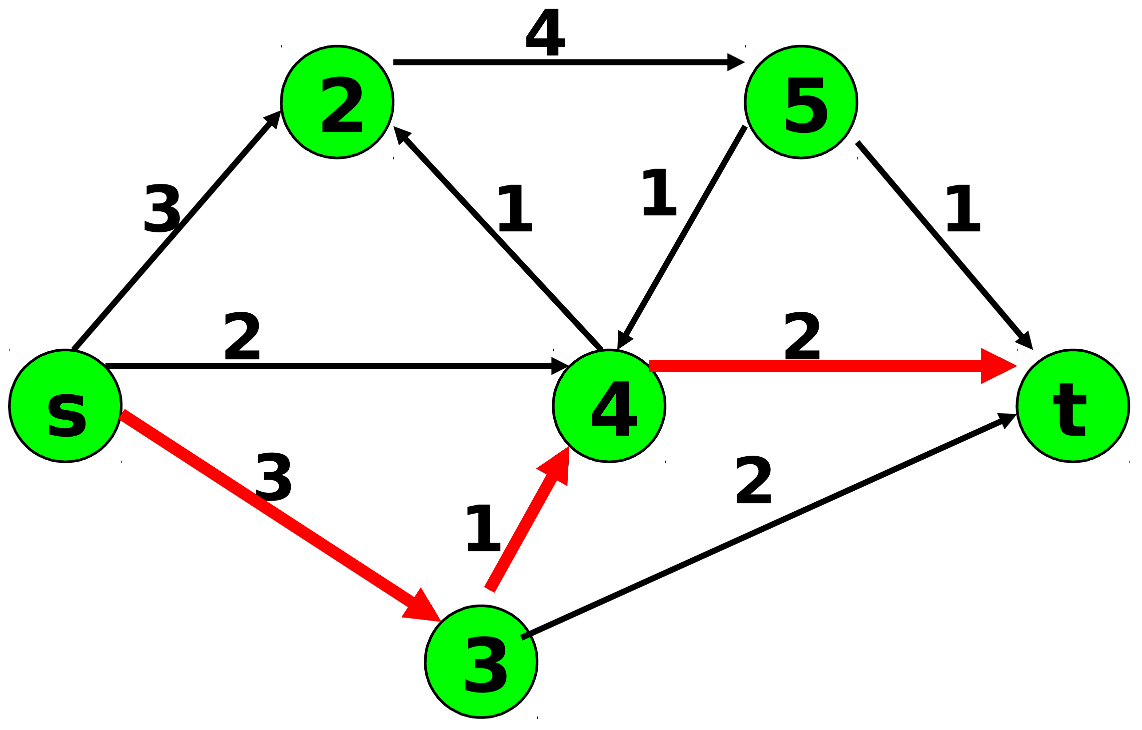
Flow 4

Ford-Fulkerson Max Flow



Flow Network

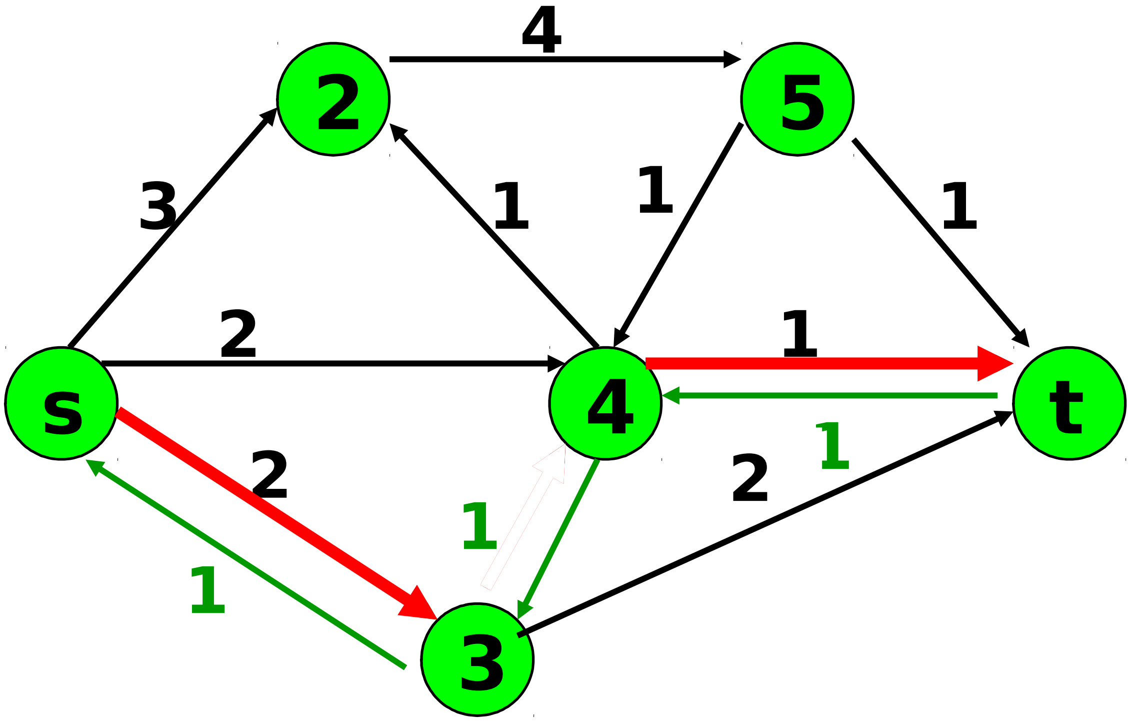
Ford-Fulkerson Max Flow



Find any s-t path in G(x)

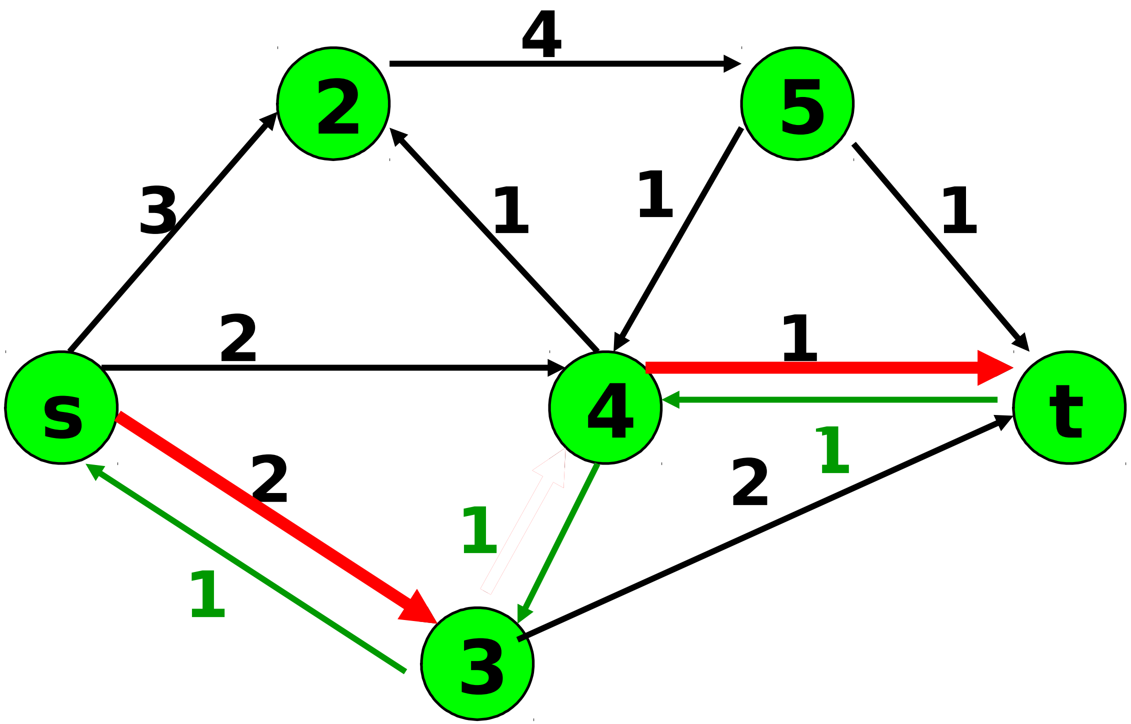
Ford-Fulkerson Max Flow

Residual Graph:

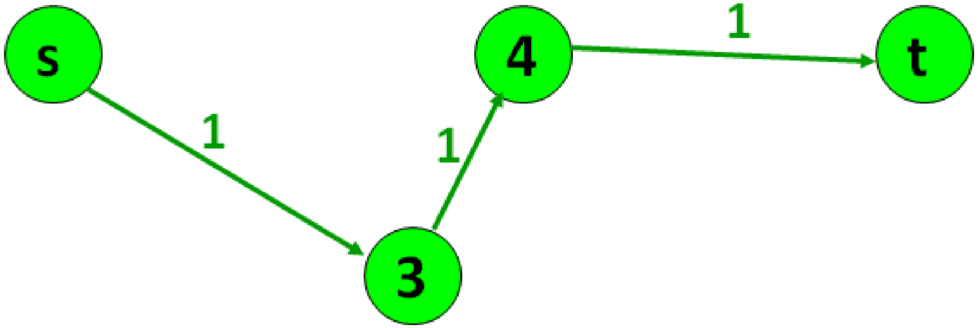


Ford-Fulkerson Max Flow

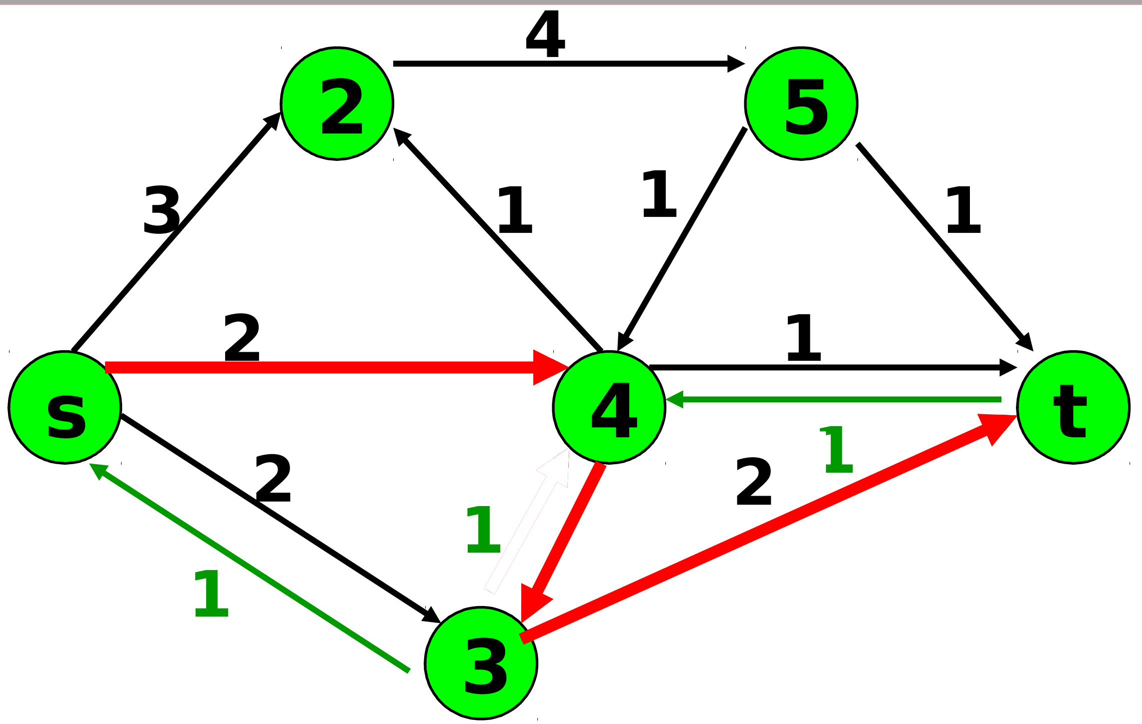
Residual Graph:



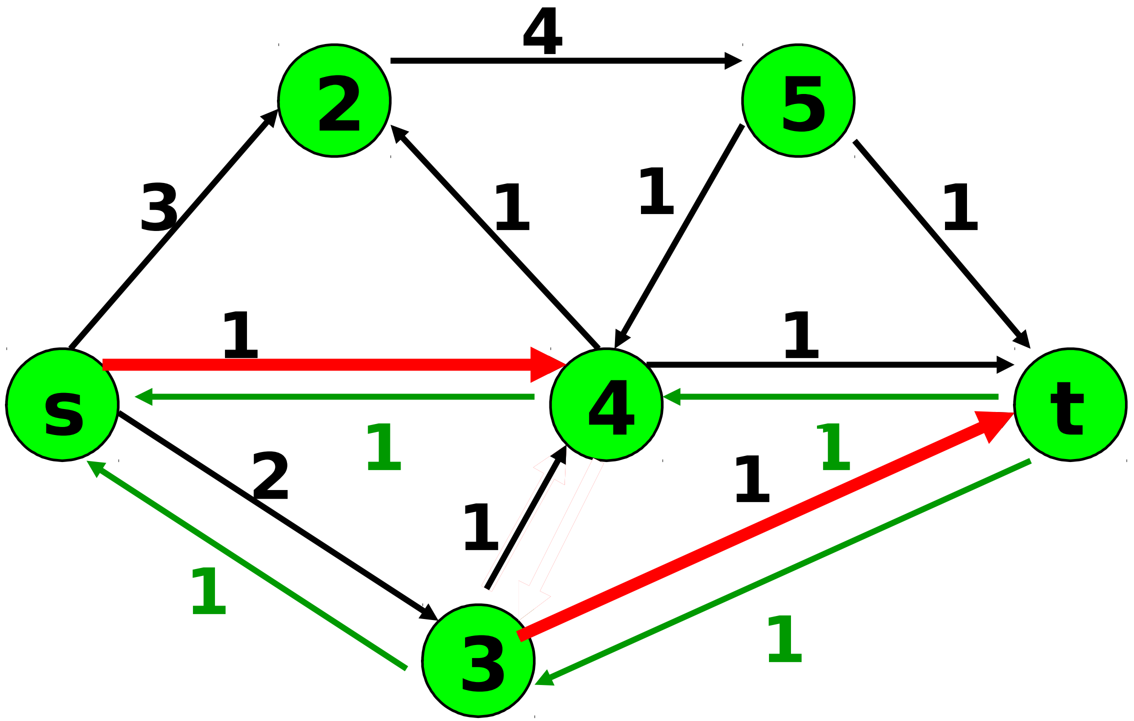
Flow:



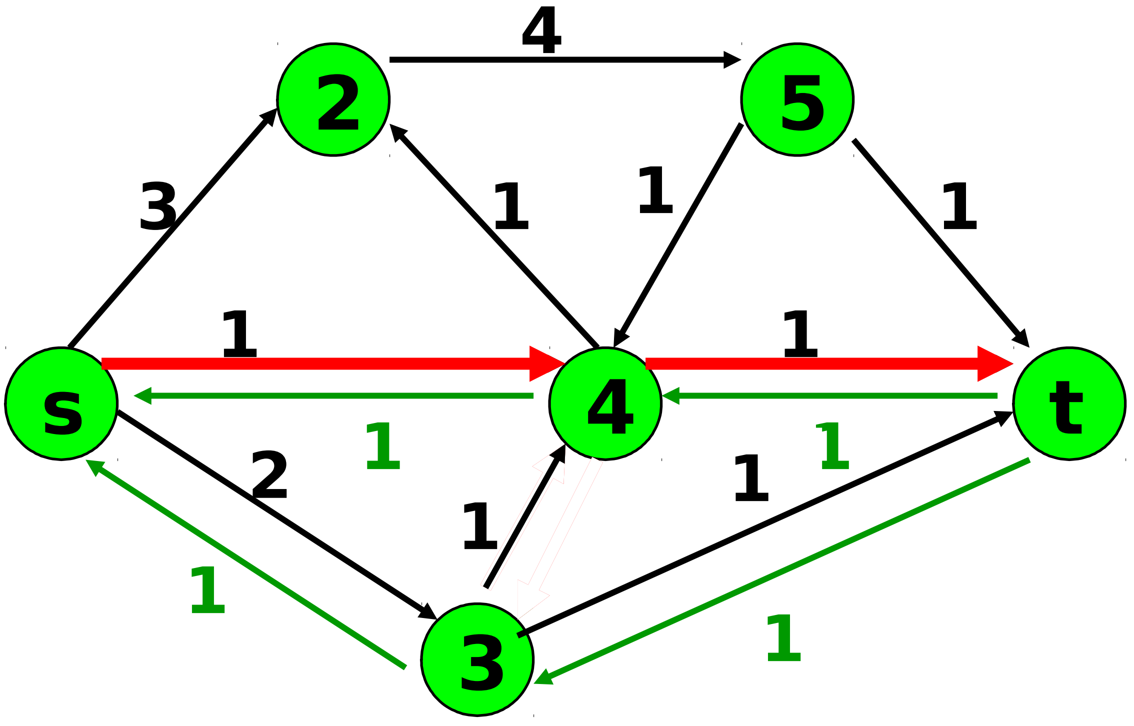
Ford-Fulkerson Max Flow



Ford-Fulkerson Max Flow

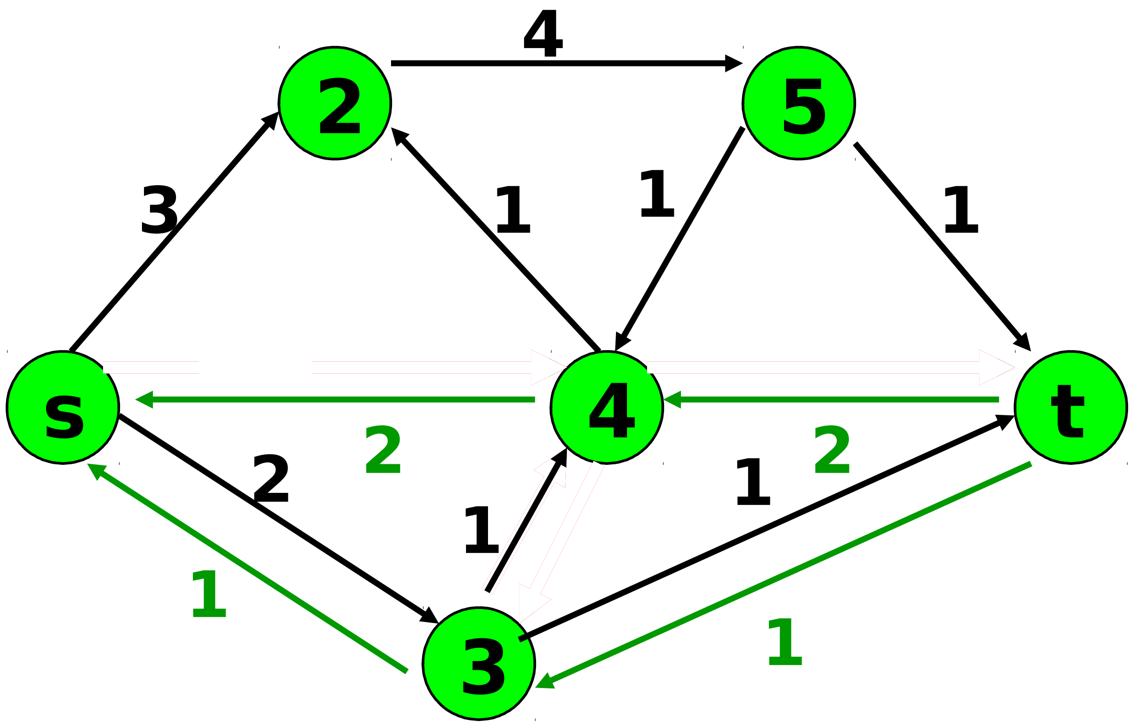


Ford-Fulkerson Max Flow



Find any s-t path

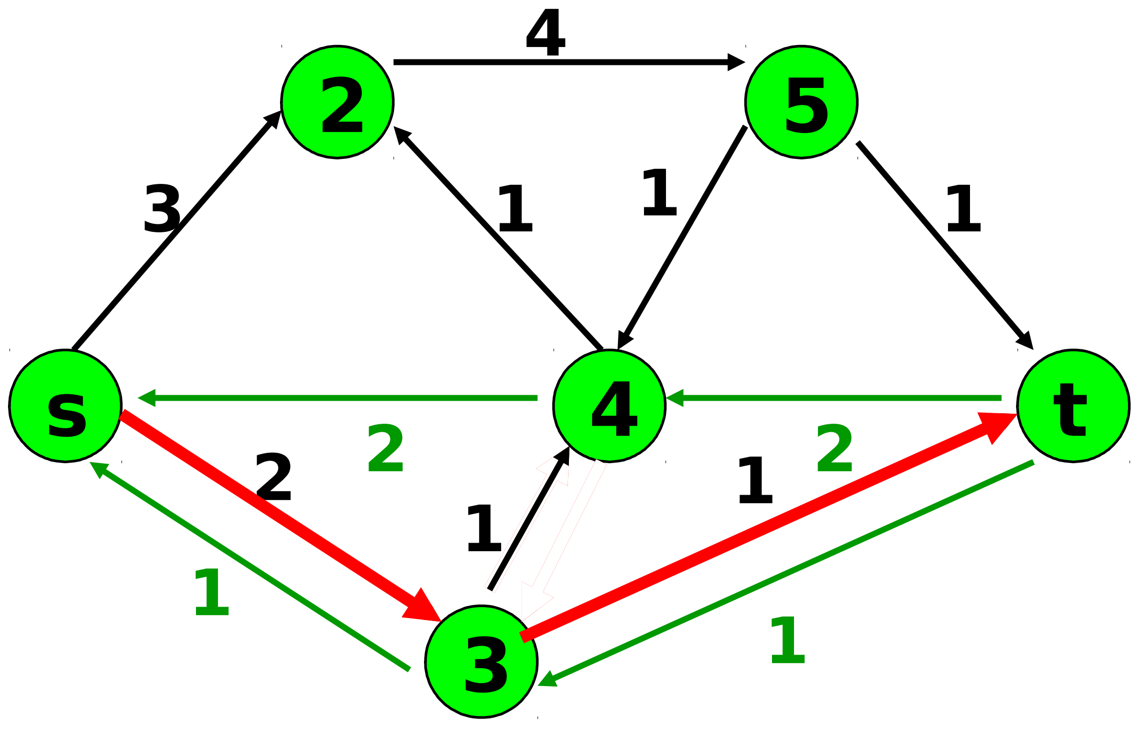
Ford-Fulkerson Max Flow



Determine the capacity Δ of the path.

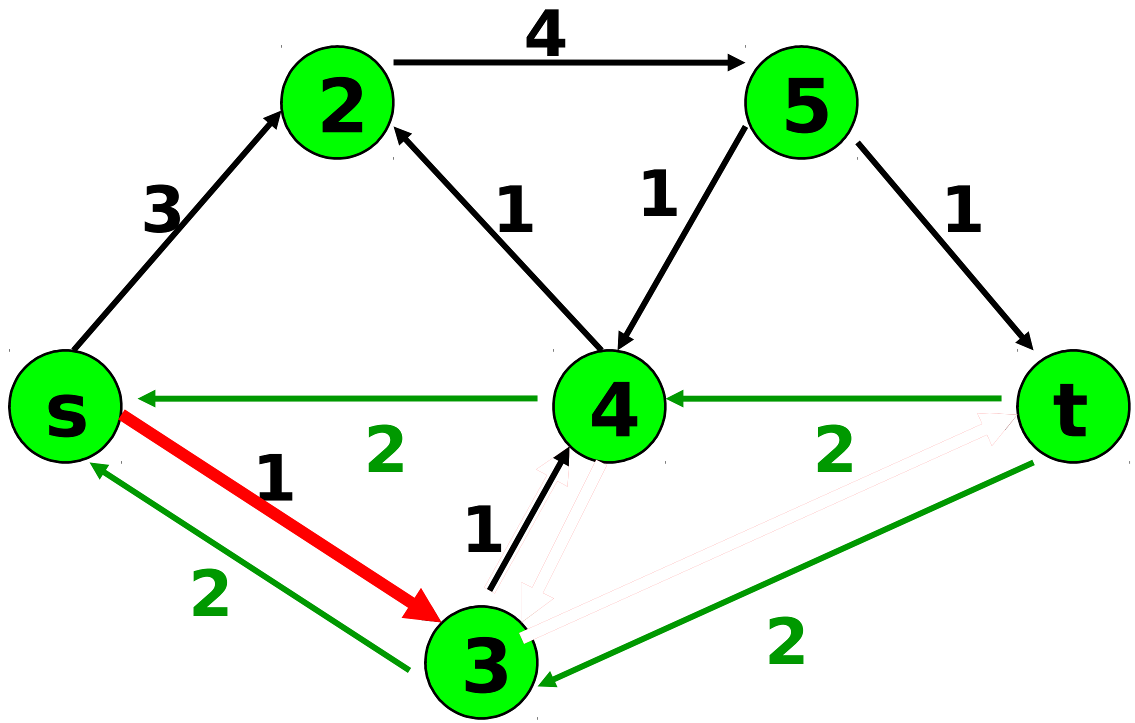
Send Δ units of flow in the path.
Update residual capacities.

Ford-Fulkerson Max Flow



Find any s-t path

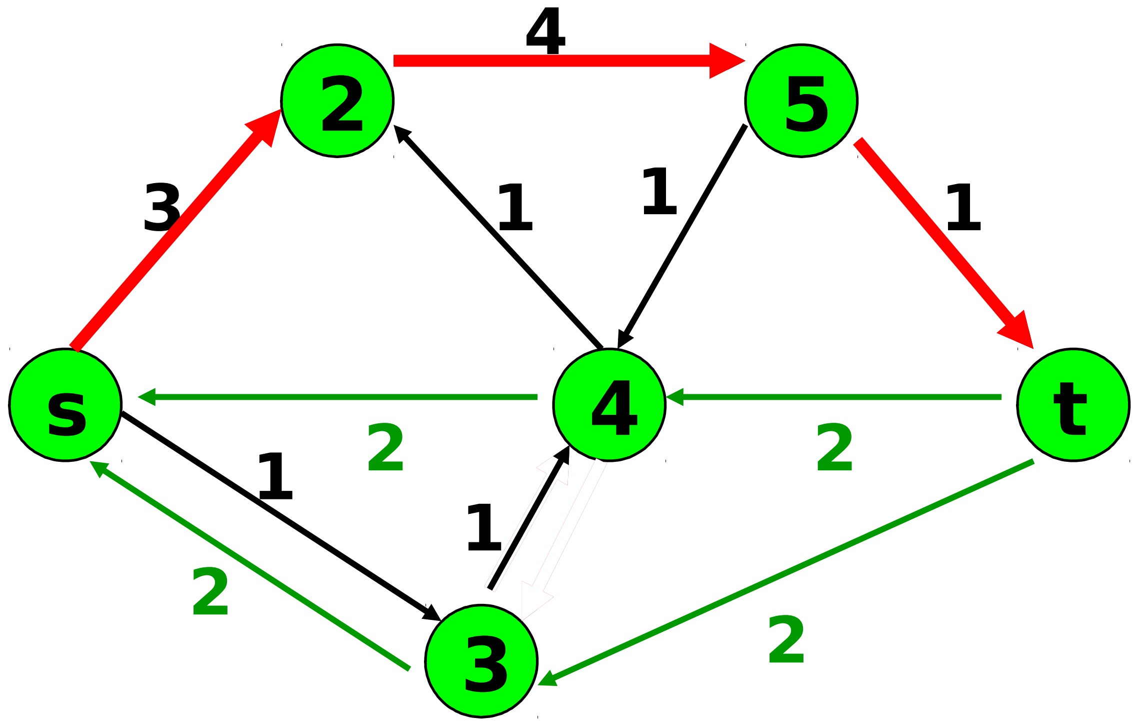
Ford-Fulkerson Max Flow



Determine the capacity Δ of the path.

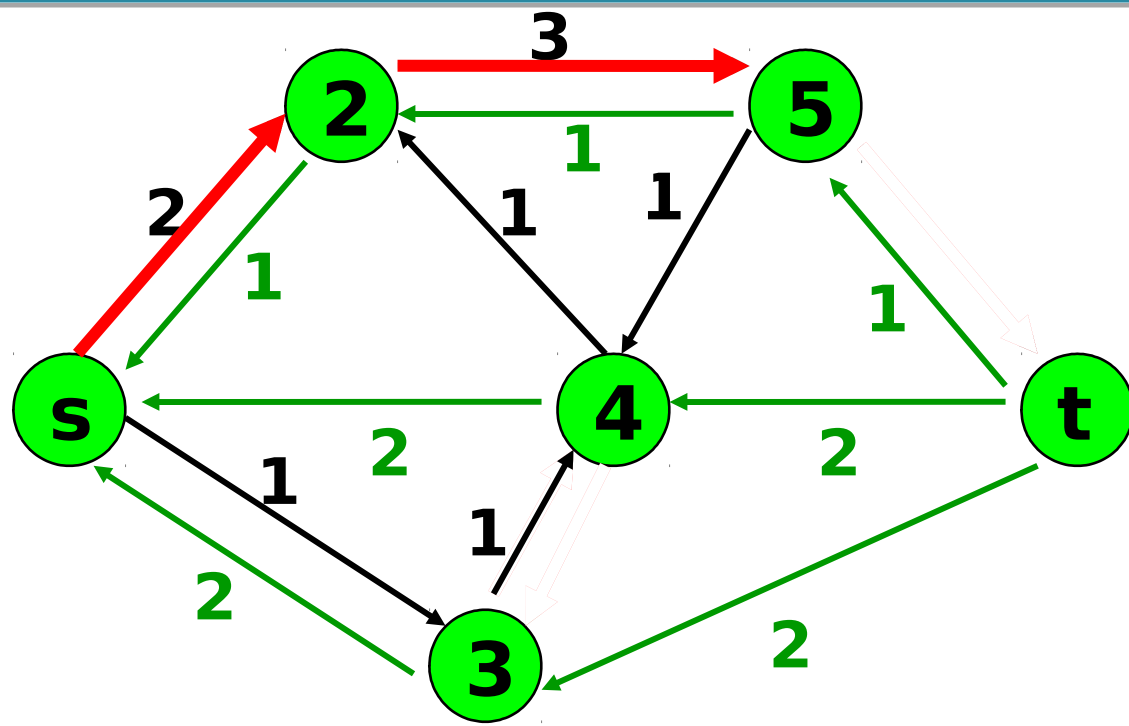
Send Δ units of flow in the path.
Update residual capacities.

Ford-Fulkerson Max Flow



Find any s-t path

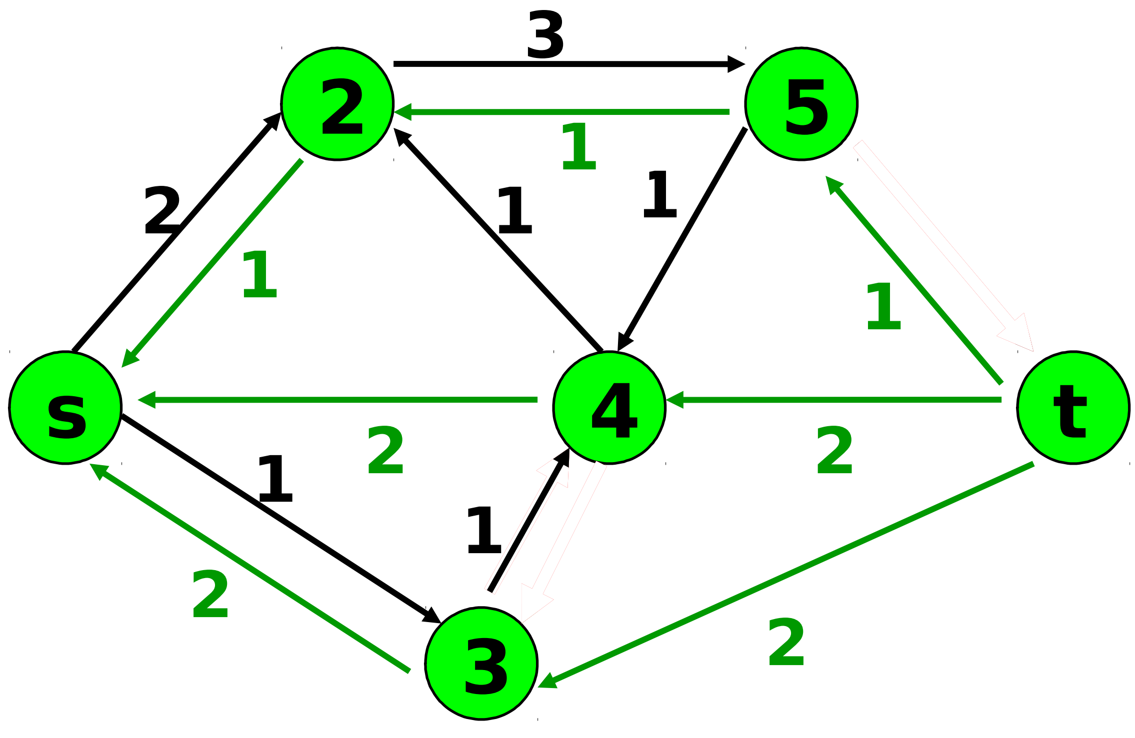
Ford-Fulkerson Max Flow



Determine the capacity Δ of the path

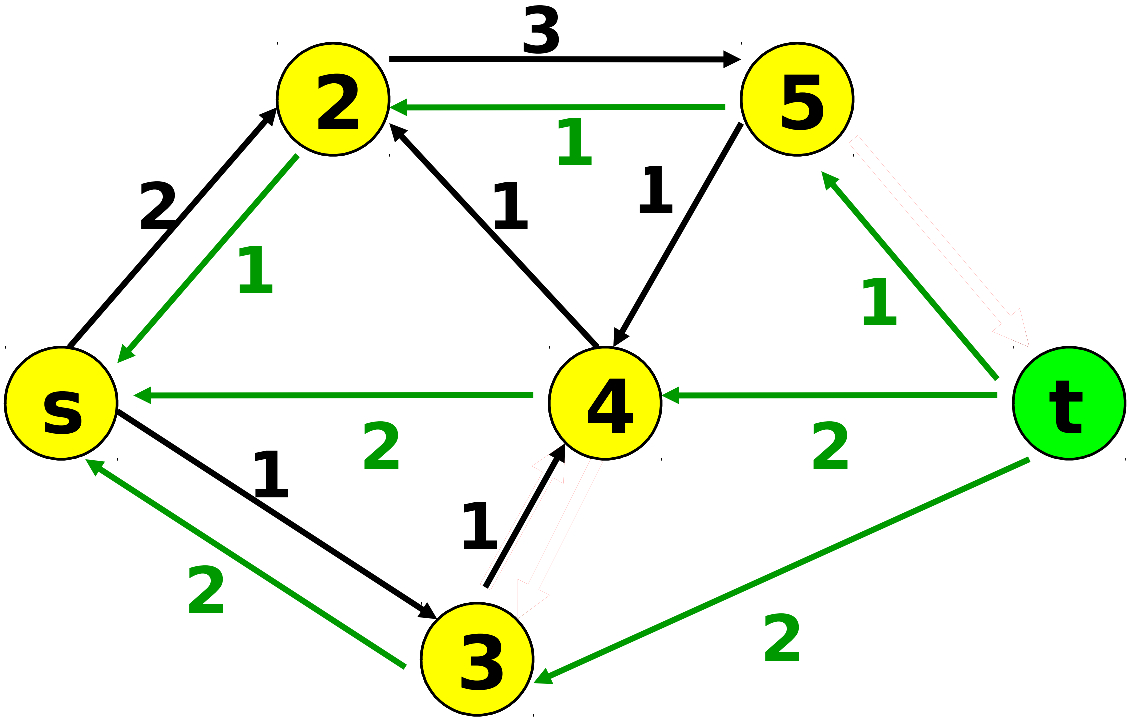
Send Δ units of flow in the path.
Update residual capacities.

Ford-Fulkerson Max Flow



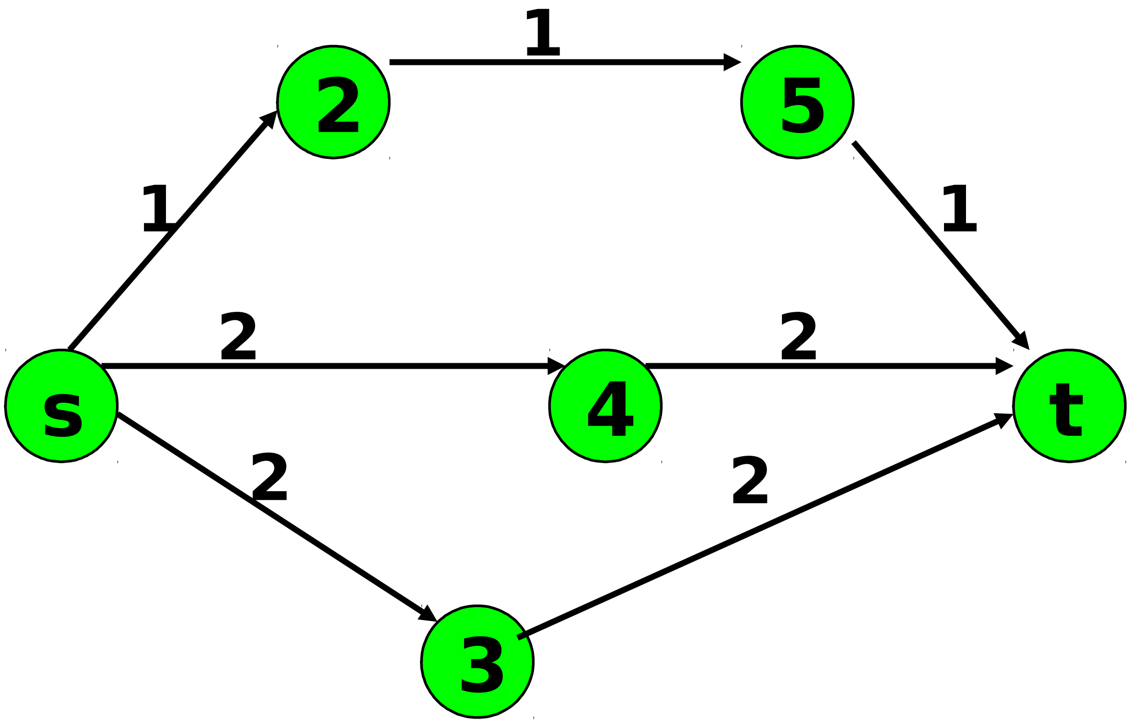
There is no s-t path in the residual network. This flow is optimal

Ford-Fulkerson Max Flow



These are the nodes that are reachable from node s.

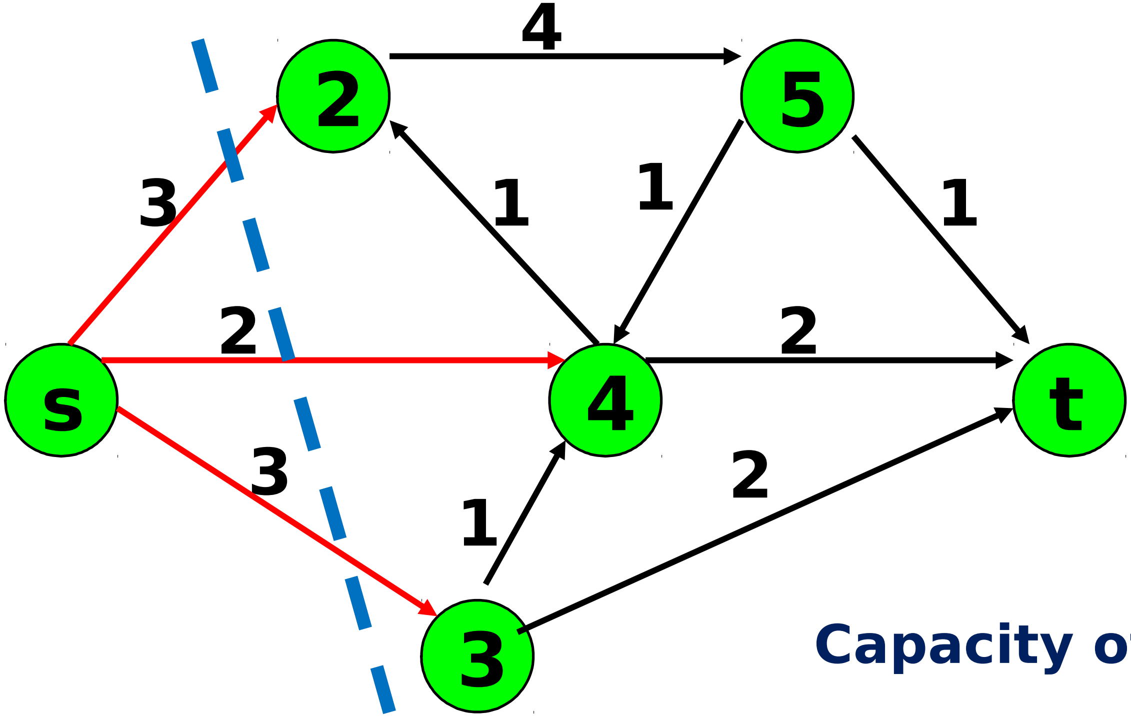
Ford-Fulkerson Max Flow



Here is the optimal flow

Max-Flow Min-Cut Theorem

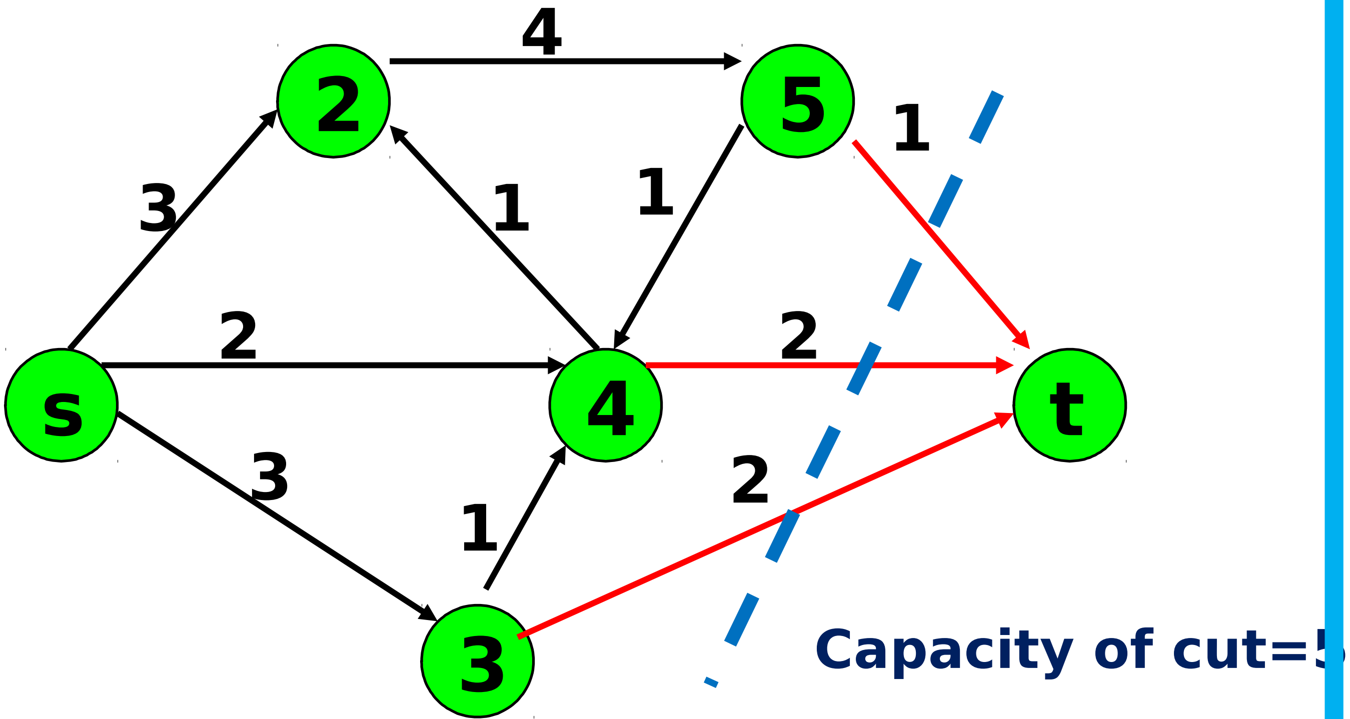
- **Definition of s-t Cut**
  - Set of edges is said to be a cut if they are removed from the graph, graph should be disconnected into two components, one with source and the other one with sink.



Capacity of cut=8

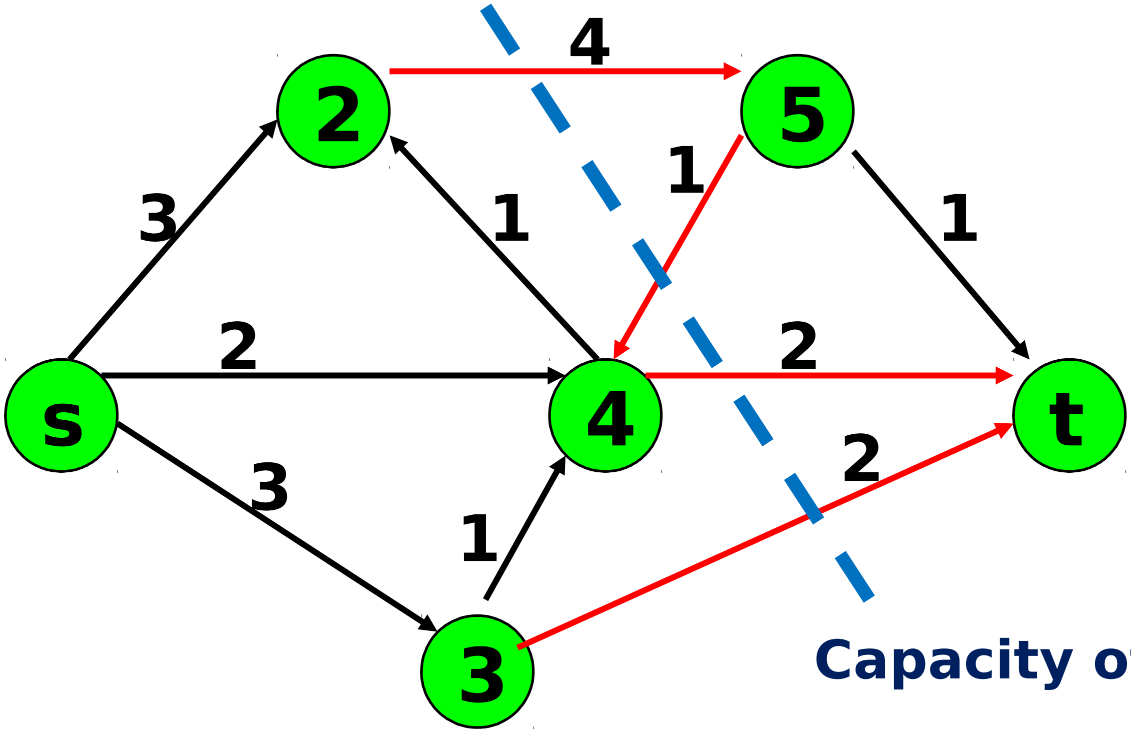
Max-Flow Min-Cut Theorem

- **Definition of s-t Cut**
  - Set of edges is said to be a cut if they are removed from the graph, graph should be disconnected two components, one with source and the other one with sink.



Max-Flow Min-Cut Theorem

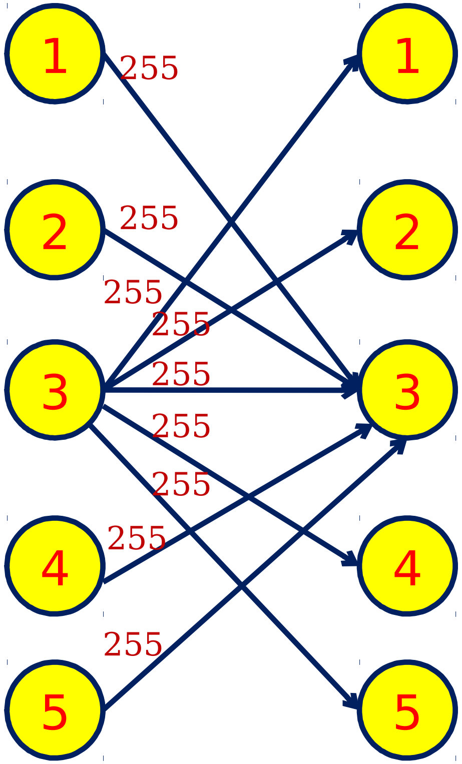
- **Definition of s-t Cut**
  - Set of edges is said to be a cut if they are removed from the graph, graph should be disconnected two components, one with source and the other one with sink.



Capacity of cut=9

Graph Vs. Image

	1	2	3	4	5
1	0	0	255	0	0
2	0	0	255	0	0
3	255	255	255	255	255
4	0	0	255	0	0
5	0	0	255	0	0



Image

Graph representation

Image Reconstruction using Max-Flow

- Consider following image

1	0	1	1	3
0	1	1	0	2
1	0	1	1	3
0	1	0	1	2
2	2	3	3	

- Given

  Row sum=3, 2, 3, 2

  Column sum=2, 2, 3, 3

- Find a binary image A such that the row sum of A is 3, 2, 3, 2 and column sum is 2, 2, 3, 3

Image Reconstruction using Max-Flow

- **Solution :**

Construct graph

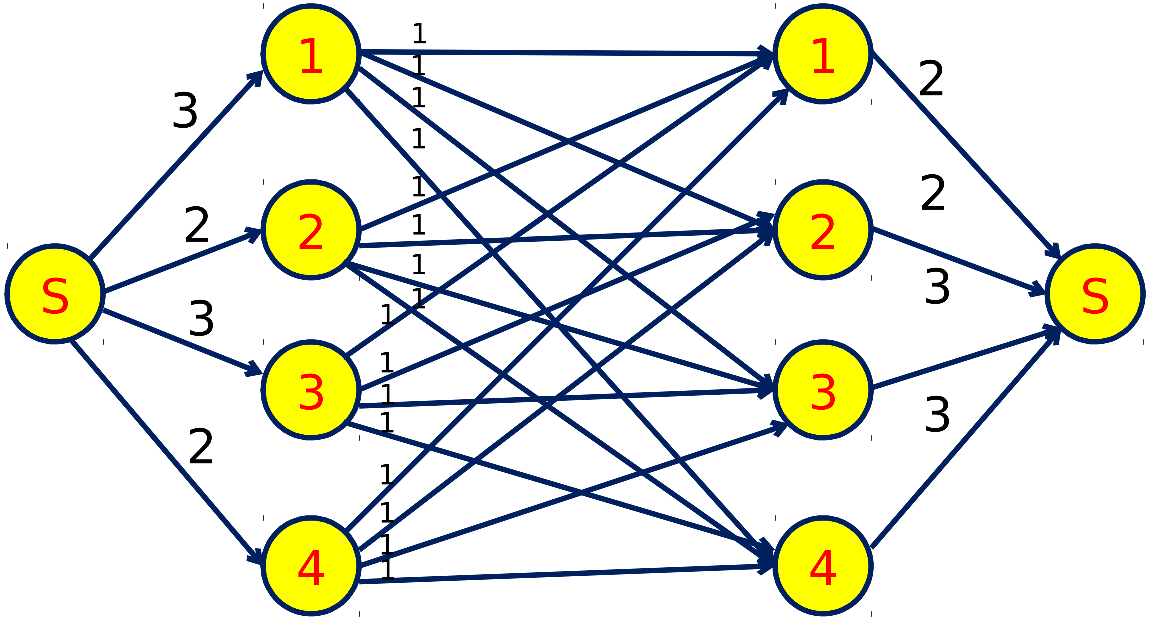
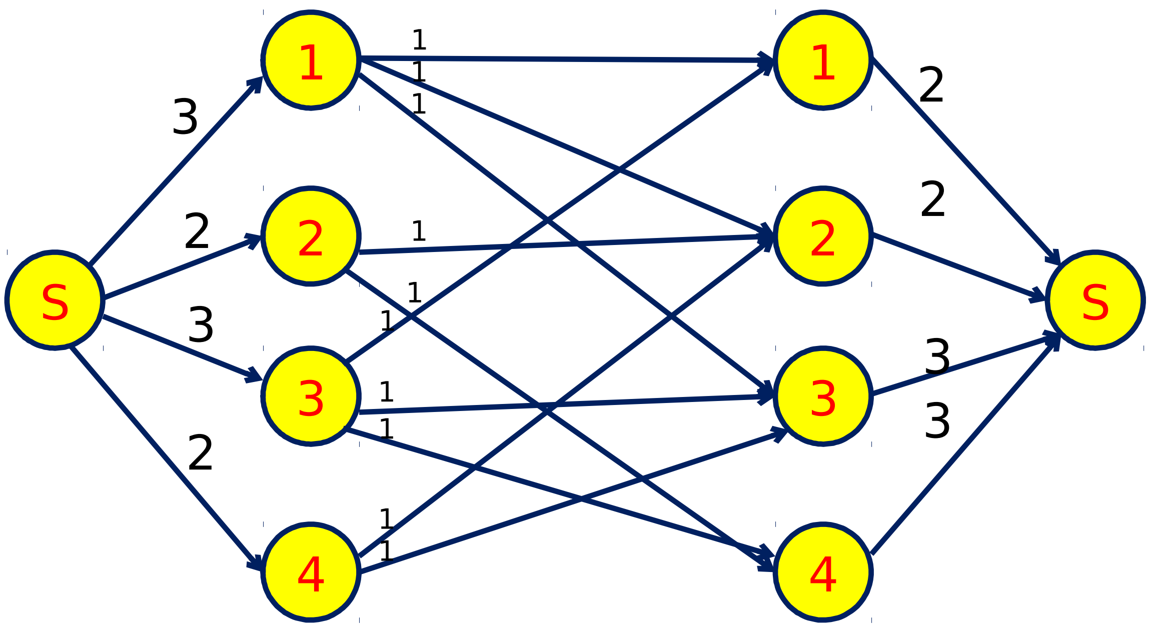


Image Reconstruction using Max-Flow

- **Max-Flow**



1	1	1	0
0	1	0	1
1	0	1	1
0	1	1	0

Reconstructed Image

Row Sum= 3, 2, 3, 2
Column Sum=2, 3, 3, 2

Image Segmentation using Min-Cut

- Image Segmentation

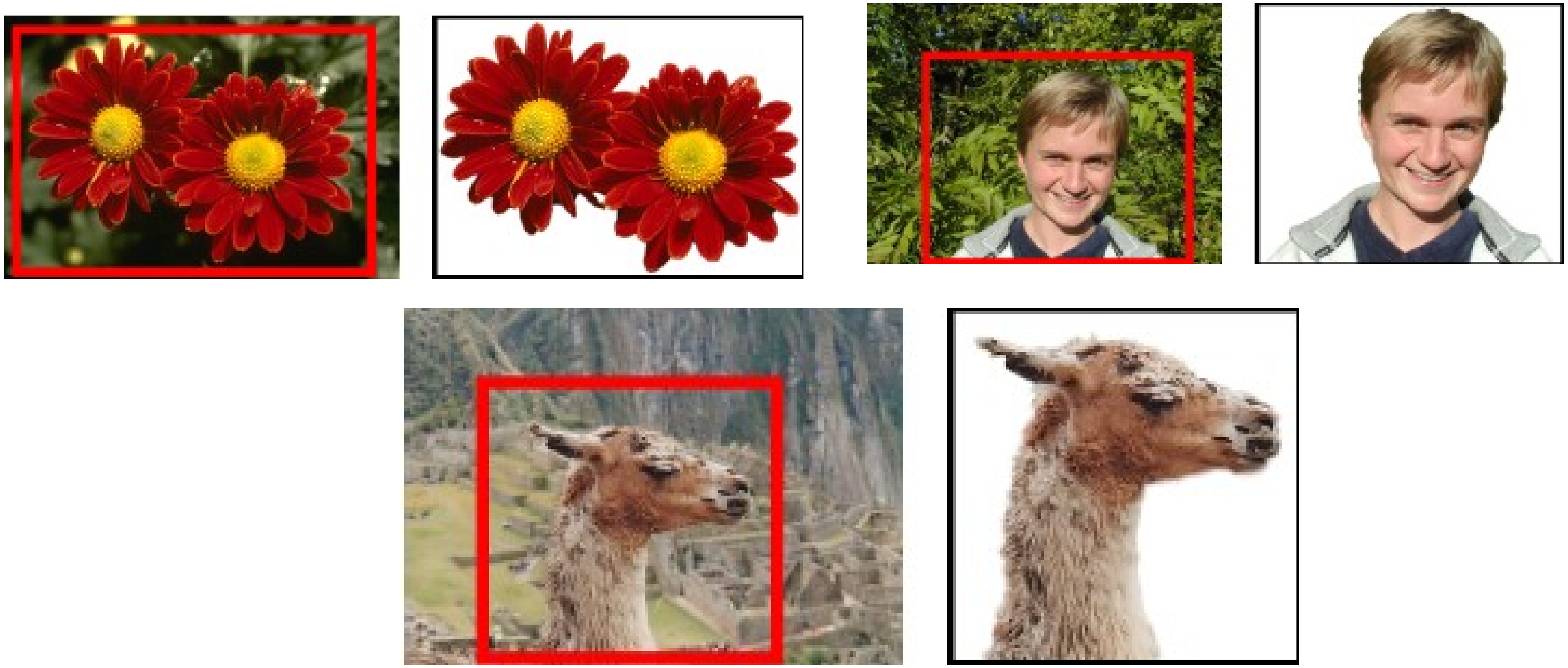


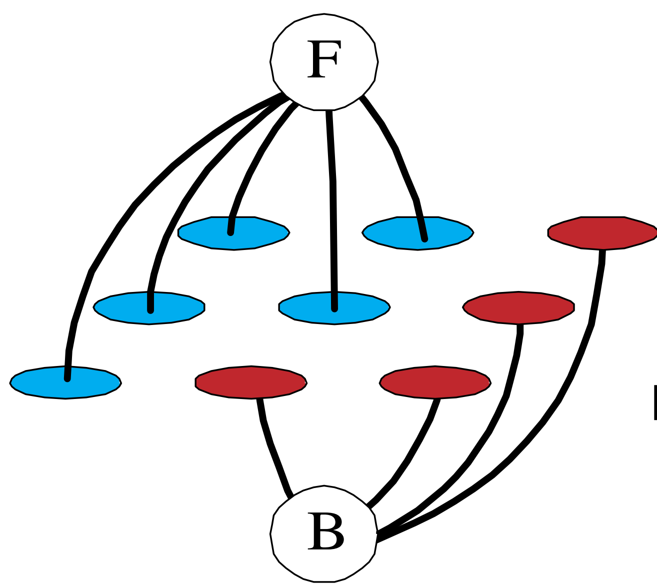
Image Segmentation using Min-Cut

- Image Segmentation

Image Segmentation using Min-Cut

- Each pixel = node
- Add two nodes F & B
- Labeling: link each pixel to either F or B

F	F	B
F	F	B
F	B	B



Desired result

Image Segmentation using Min-Cut

- **Construct graph with data term**
  - Put one edge between each pixel and F
  - Put one edge between each pixel and B
  - Weight of edge between i and F: $w_{iF} = -\lambda \log(P_B(i))$
  - Weight of edge between i and B: $w_{iB} = -\lambda \log(P_F(i))$

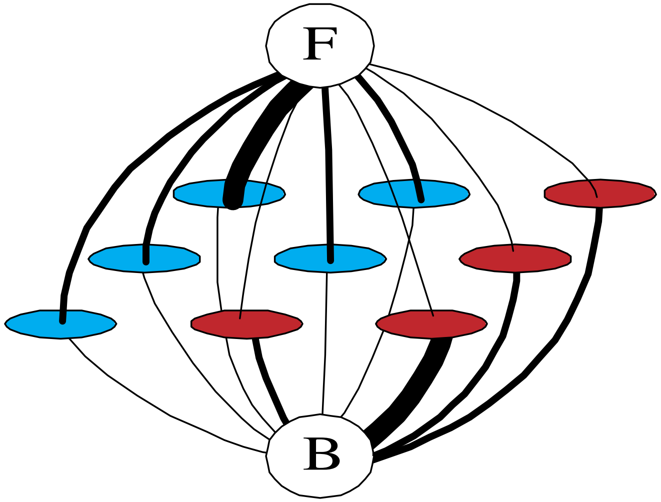


Image Segmentation using Min-Cut

- **Add Smoothness term to the Graph**
  - Add an edge between each neighbor pair (i,j)
  - Weight of edge between i and j: $w_{ij} = \exp(-(I_i - I_j)^2 / 2\sigma^2)$

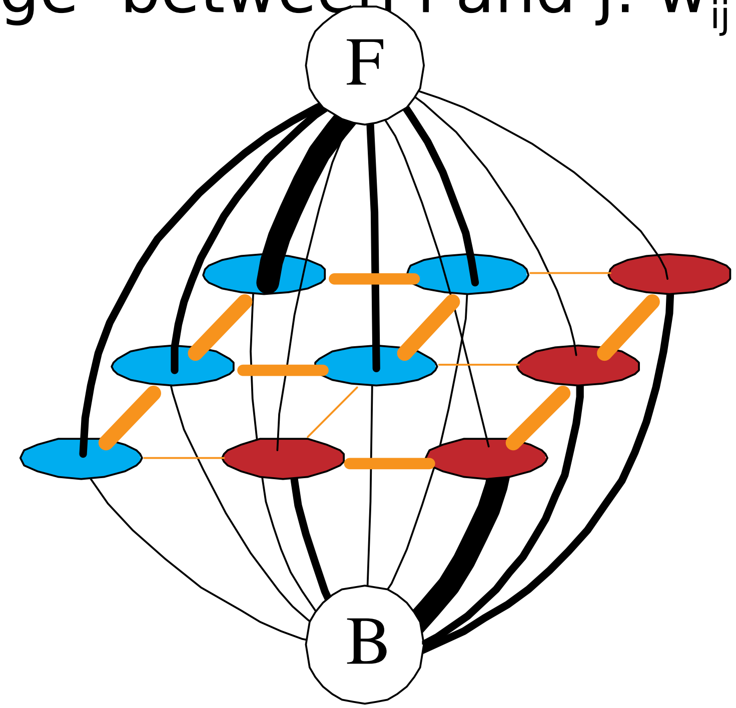


Image Segmentation using Min-Cut

- **Min-cut**
  - Cut: Remove edges to disconnect F from B
  - Min Cut: Cut with sum of its edge weights is minimum

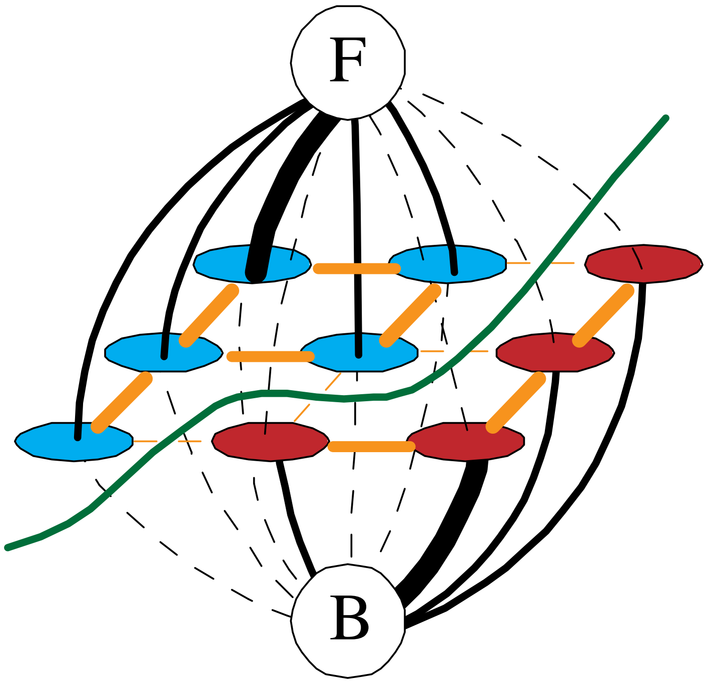


Image Segmentation using Min-Cut

- In order to be a cut:
  - For each pixel, either the F or G edge has to be cut
- In order to be minimal
  - Only one edge to F or B per pixel can be cut

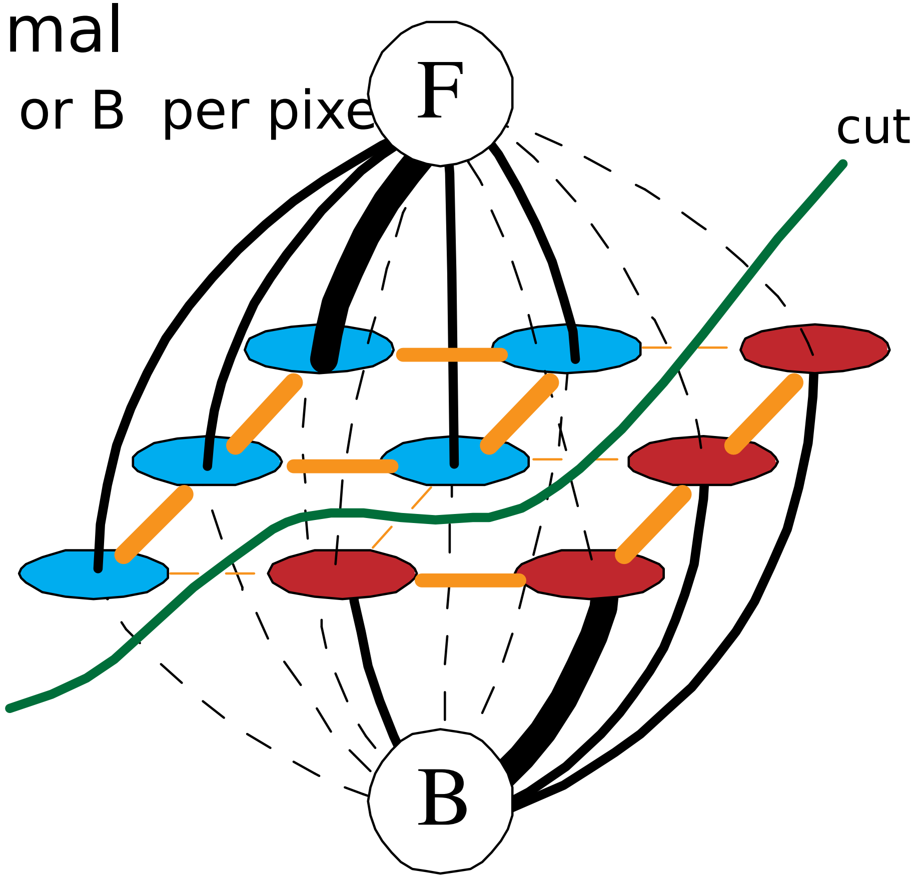


Image Segmentation using Min-Cut

- Which edges are to be removed?
  - Edges with least cost

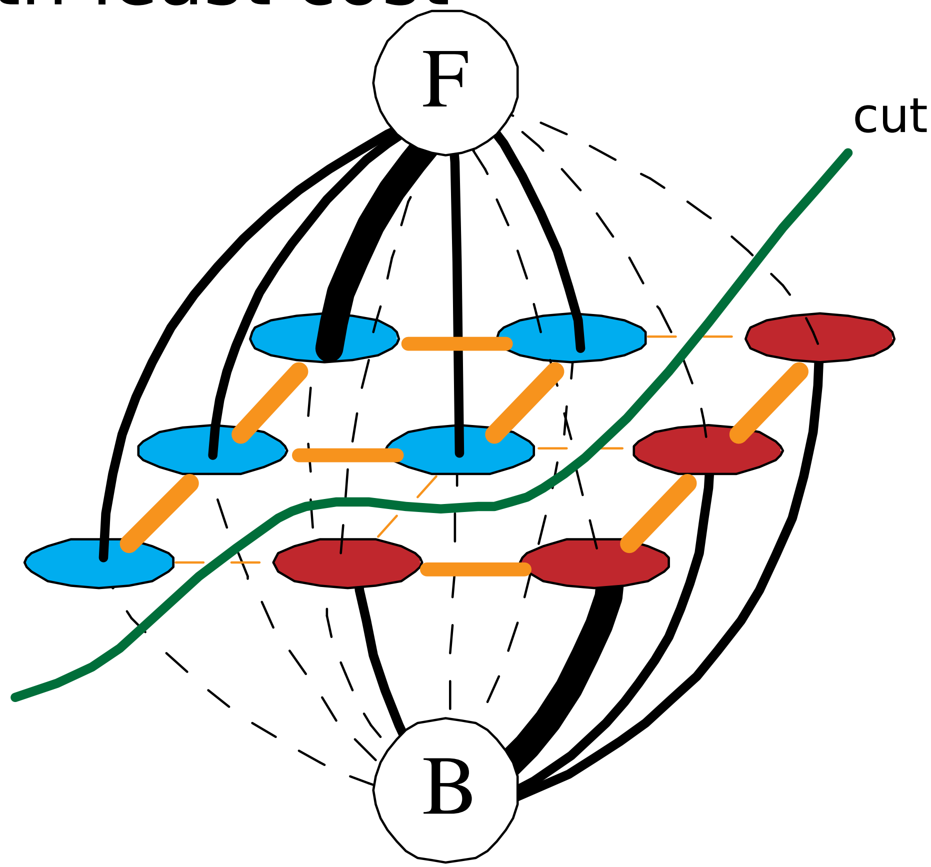


Image Segmentation using Min-Cut

- Since $w_{iF} = -\lambda \log(P_B(i))$,

  larger value of $P_B(i)$ will lead to smaller value of w_{iF}

  - Hence edge (i,F) will be removed, and edge (i,B) will stay
  - ie. Edge (i,B) will stay after cut if prob of i to be background is high

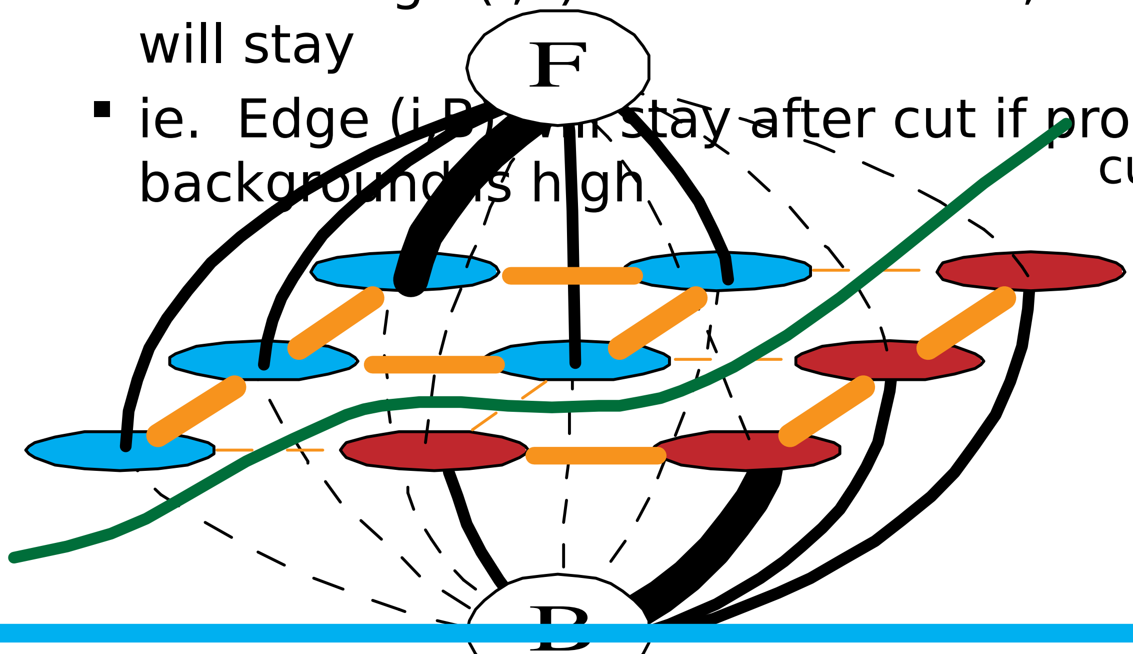


Image Segmentation using Min-Cut

- Since $w_{iB} = -\lambda\log(P_F(i))$,

  larger value of $P_F(i)$ will lead to smaller value of w_{iB}

  - Hence edge (i,B) will be removed, and edge (i,F) will stay
  - ie. Edge (I,F) w y after cut if prob of i to be foreground is h

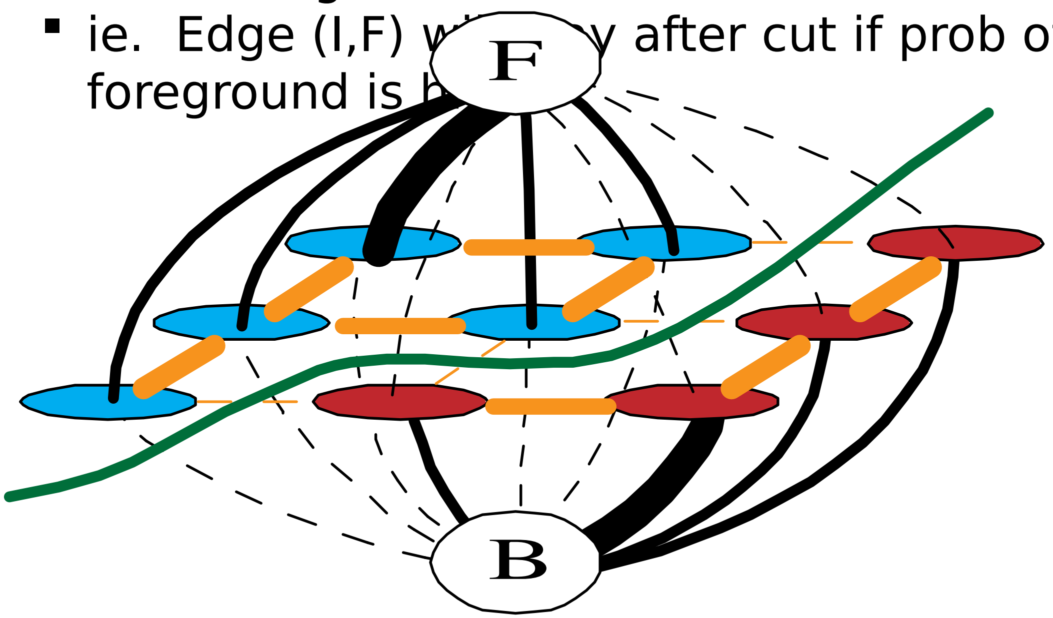


Image Segmentation using Min-Cut

- Since for neighboring pixels (i,j), $w_{ij} = \exp(-(I_i - I_j)^2 / 2\sigma^2)$, similar i and j, w_{ij} is very high.
  - Hence edge between similar pixels will stay after graph cut.
  - This provides smoothness

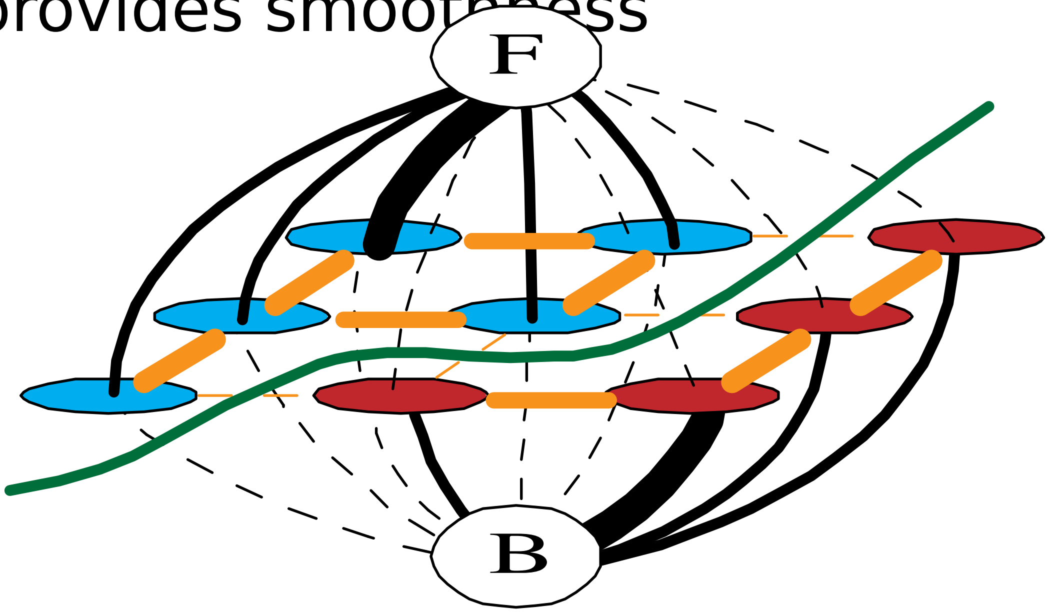


Image Segmentation using Min-Cut

- **How to find min-cut**
  - Apply max-flow algorithm
  - The output of max-flow algorithm will result in min-cut (max-flow min-cut theorem)

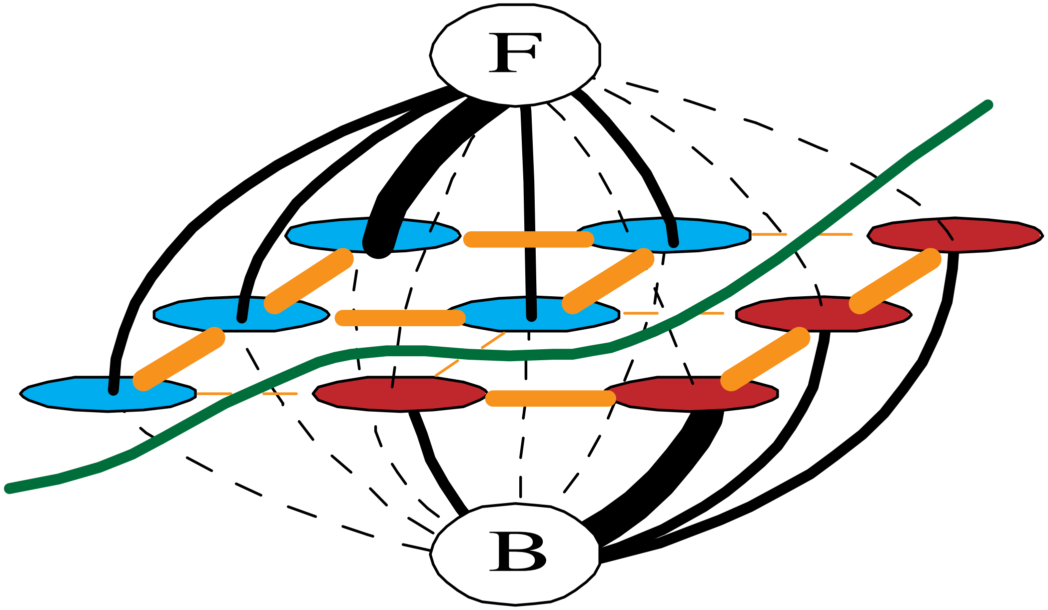


Image Segmentation using Min-Cut

- **How to find P_B (i) and P_F (i) for pixel I?**
  - User will give some seed of the background and foreground
  - From seed points of background compute P_B(i)
  - From seed points of foreground compute P_F(i)

Summary

- Ford-Fulkerson algorithm to find max-flow is discussed
- Relationship between image and graph is discussed
- Graph-cut was computed using Ford-Fulkerson
- Couple of applications for graph-cut were illustrated
- As computation of graph-cut algorithm is polynomial,

the applications discussed require only polynomial time.

???....

Thank You…