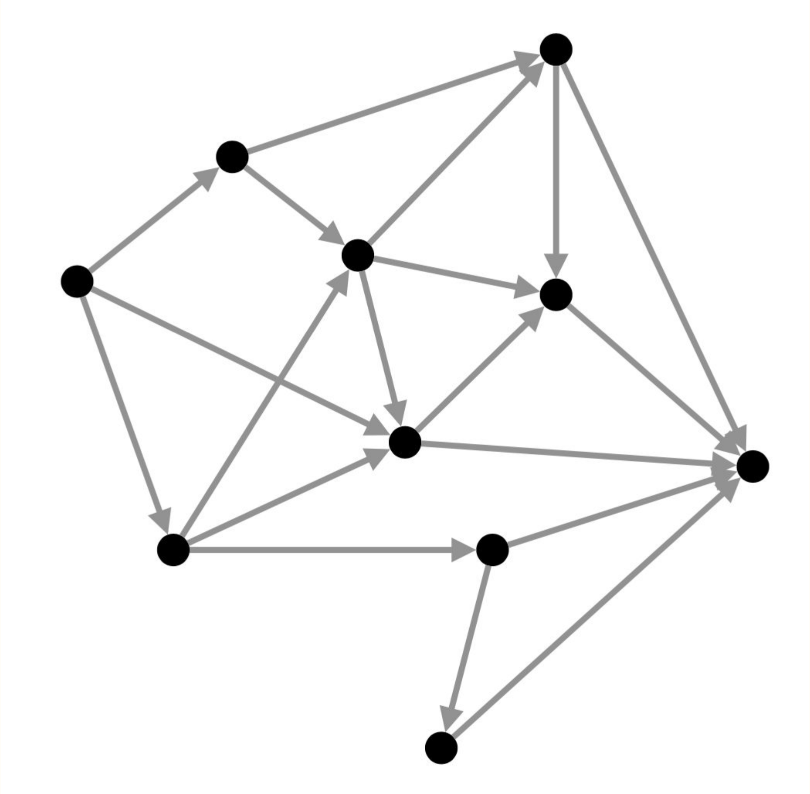
Data Structures and Algorithms

Lab 13
Maximum flow. Minimum cut

Flow networks

- What is a flow network?



Flow networks

A Flow network is a directed graph where each edge has a capacity and a flow. They are typically used to model problems involving the transport of items between locations, using a network of routes with limited capacity.

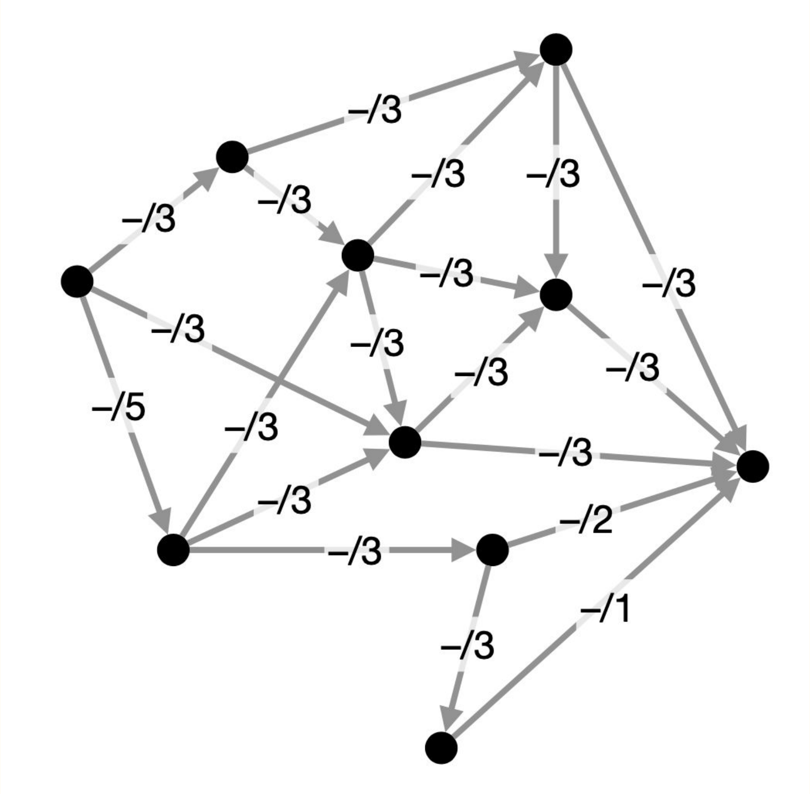
Examples include modeling traffic on a network of roads, fluid in a network of pipes, and electricity in a network of circuit components.

A flow network N is a tuple $N = (G, c, s, t)$ where

$G = (V, E)$ is a directed graph of vertices V and directed edges E
$c : E \to \mathbb{R}_0^+$ is a mapping from the edges to the nonnegative reals, and $c(e)$ is called the **capacity** of edge e
$s, t \in V$ are special vertices of G called the source and sink, respectively

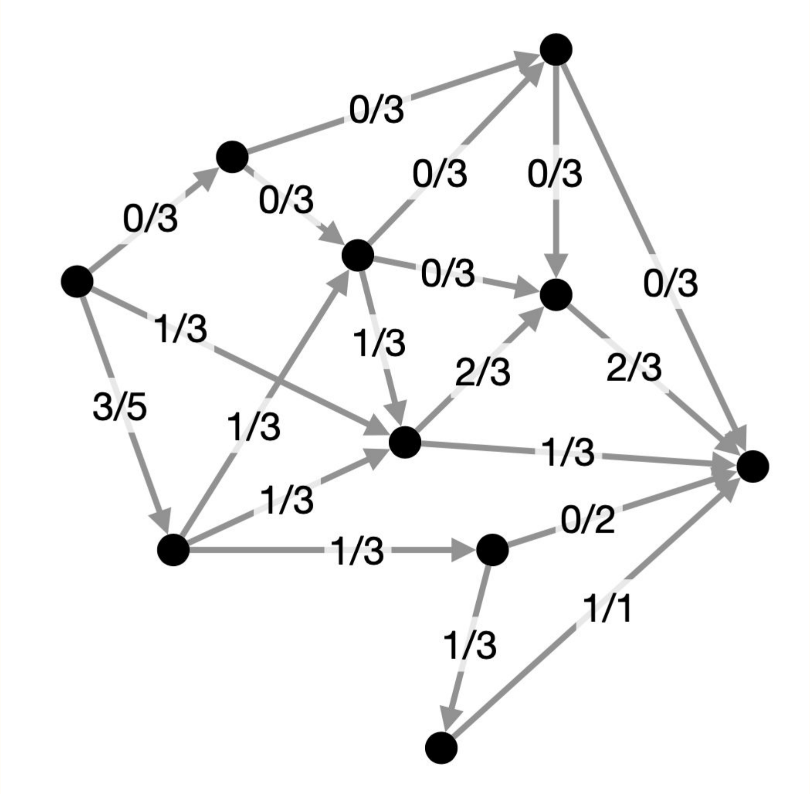
Flow networks

- What is a flow network?



Flow networks

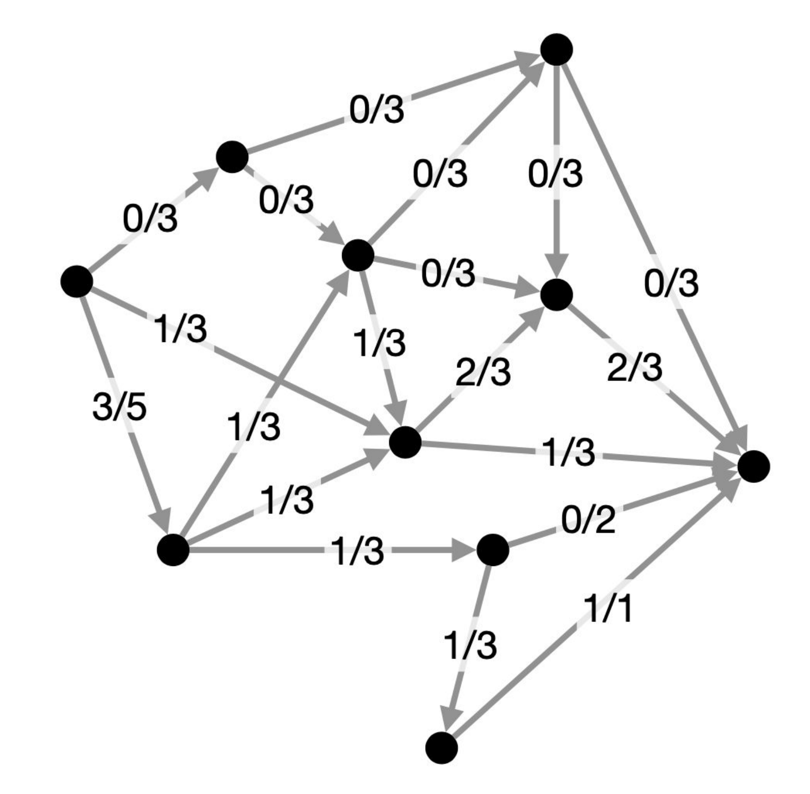
- What is a flow network?
- What is a flow?



Flow networks

- What is a flow?
- A *flow* in G is a real-valued function $f : V \times V \rightarrow R$ that satisfies the following two properties:
  - **feasibility condition**
    - “flow from u to v should be non-negative and within the capacity”
  - **flow conservation condition**
    - “flow in equals flow out”
- The **value** of the flow

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



Flow networks

A flow f for a flow network N is a mapping $f: E \to \mathbb{R}_0^+$ satisfying the following two constraints:

$$0 \leq f(e) \leq c(e)$$

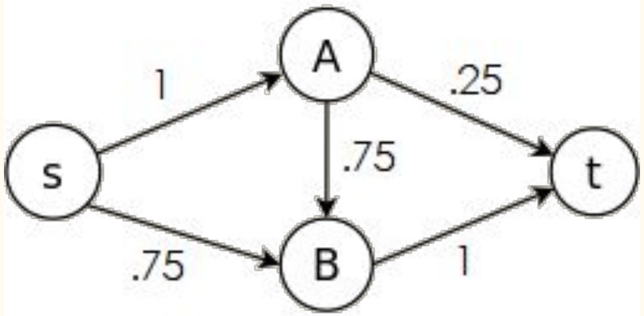
for all edges $e \in E$

$$\sum_{e^+=v} f(e) = \sum_{e^-=v} f(e)$$

for each vertex $v \neq s, t$, where e^- and e^+ denote the start and end vertex of edge e, respectively

Flow networks

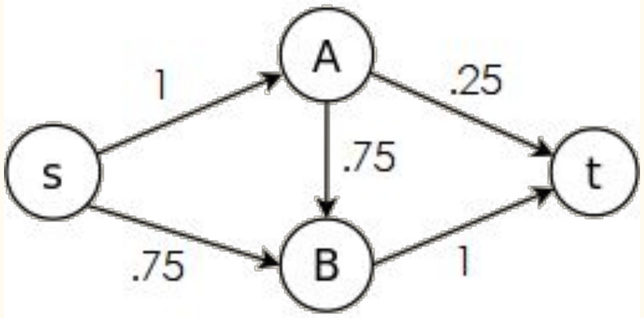
Given the flow network shown here, is the mapping (with each edge label equal to that edge's flow) an admissible flow?



Flow networks

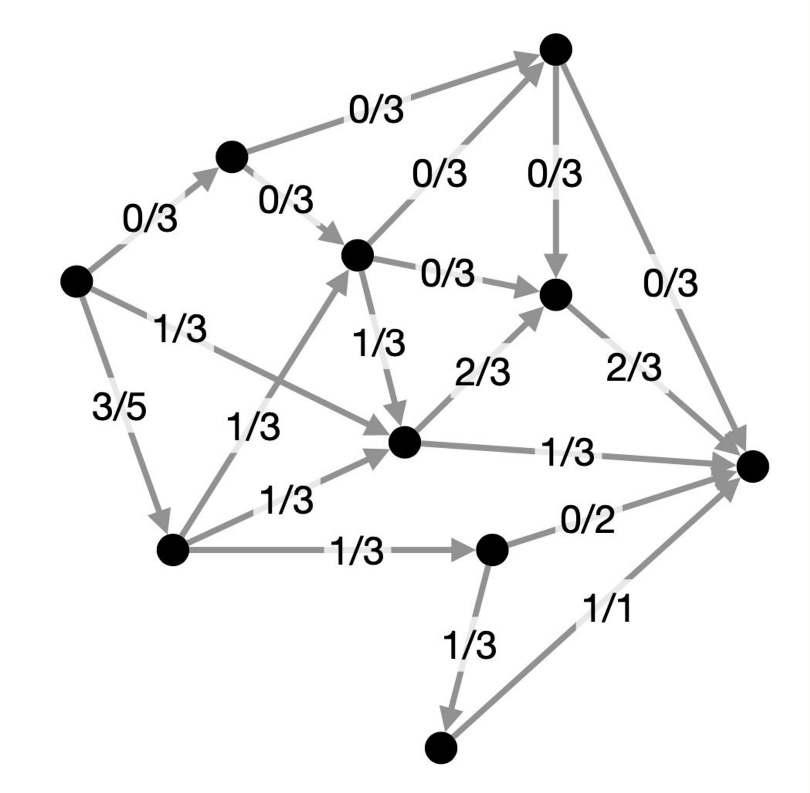
No, the mapping is not an admissible flow. While the mapping satisfies the feasibility condition

The flow conservation condition is not satisfied. Specifically, the total flow exiting node b is 1, while the total flow entering node b is 1.5



Flow networks

- What is a flow network?
- What is max flow?
- How to increase flow?
- How do you know if it is possible to increase flow?



Max flow = ?

Flow networks

- The *maximum flow* problem is the problem of finding the maximum admissible flow through a single source, single sink flow network.
- It was originally formulated in 1954 by mathematicians attempting to model Soviet railway traffic flow.
- Well known solutions for the maximum flow problem include the Ford-Fulkerson algorithm.
- Examples of applications include
  - airline flight crew scheduling,
  - the circulation-demand problem (where goods with location dependent demand must be transported using routes with limited capacity)

Flow networks

The residual capacity for a flow network N and flow f is a mapping $r : E \to \mathbb{R}_0^+$ such that

$$r(e) = c(e) - f(e)$$

for all edges $e \in E$

An augmenting path for a flow network N and flow f is a directed path p starting at the source s and ending at the sink t such that the residual capacity r for all edges on that path is nonzero. Specifically,

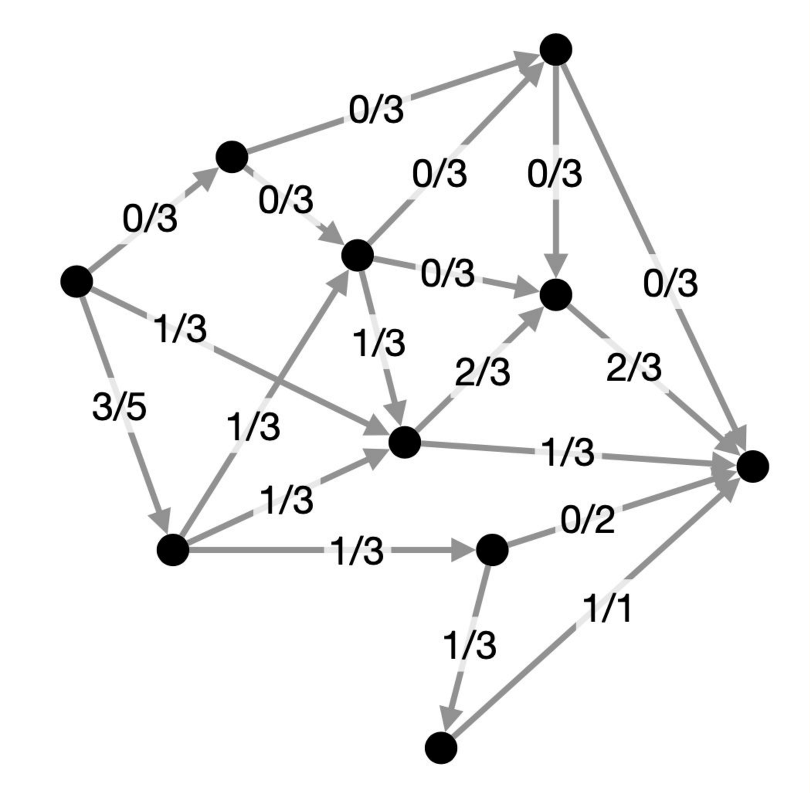
$p = (u_1, u_2, \ldots, u_{k-1}, u_k)$ is a directed path with $u_1 = s$ and $u_k = t$ such that

$$r(e) = c(e) - f(e) > 0$$

for all edges $e = (u_i, u_{i+1})$ for $i = 1, \ldots, k-1$ in the path p.

Flow networks

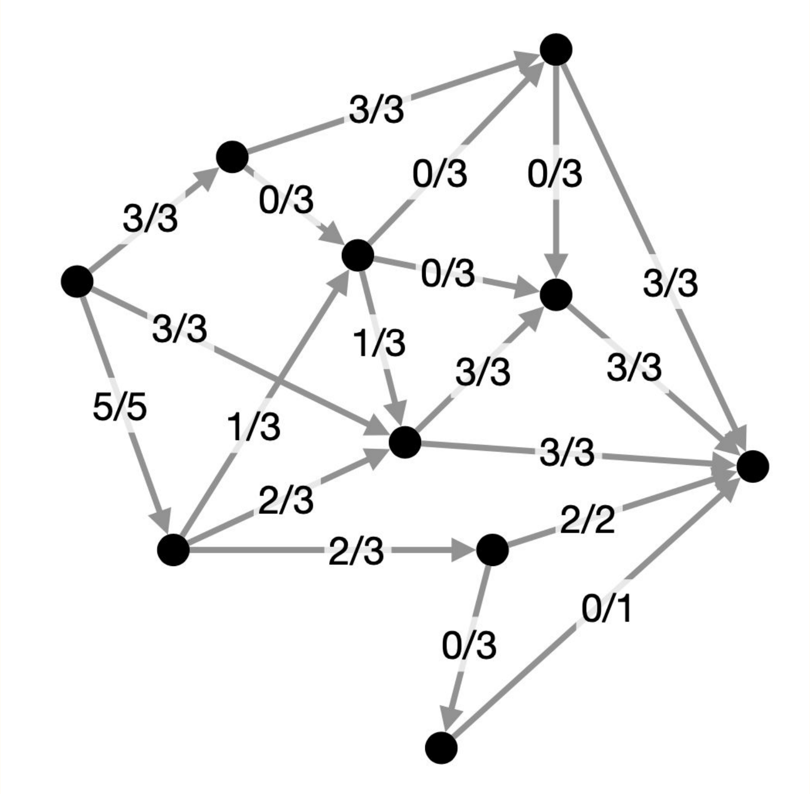
- What is a flow network?
- What is max flow problem?
  - we are given a flow network G with source s and sink t, and we wish to find **a flow of maximum value**.
- How to increase flow?
  - Ford-Fulkerson method.
  - By finding an “augmenting path” in an associated “residual network” G_f
- How do you know if it is possible to increase flow?
  - The residual network has augmenting paths.



Max flow = ?

Flow networks

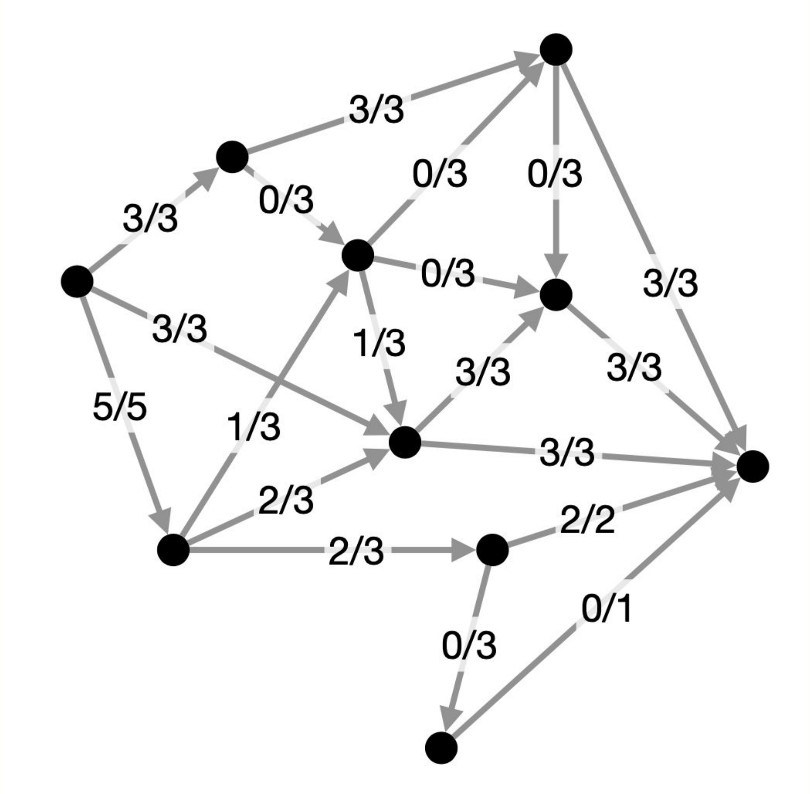
- What is a flow network?
- What is max flow?
- How to increase flow?
- How do you know if it is possible to increase flow?



Max flow = 11

Flow networks

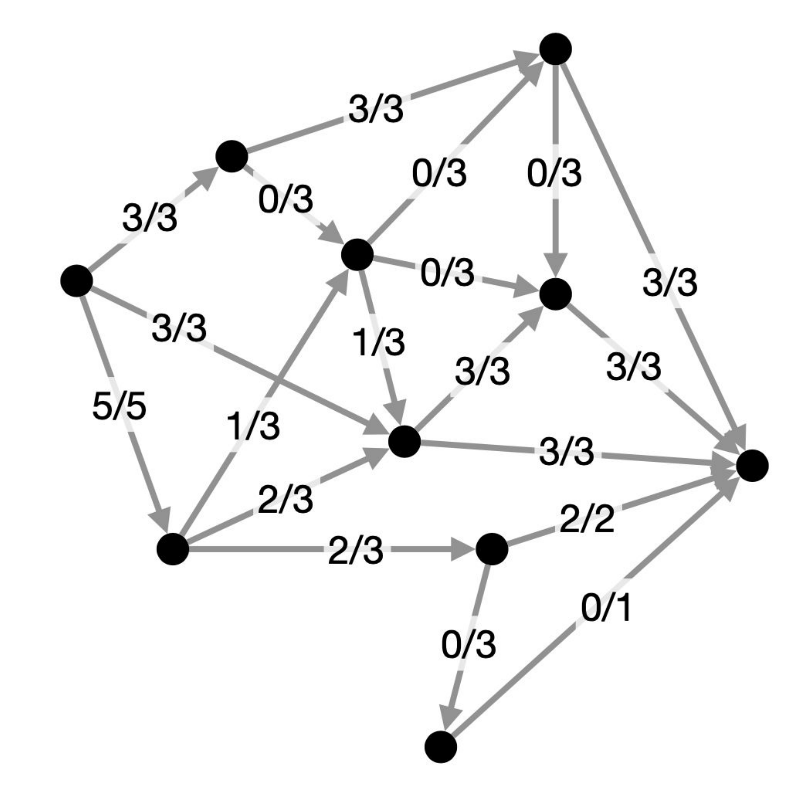
- What is a flow network?
- What is max flow?
- How to increase flow?
- How do you know if it is possible to increase flow?
- How to prove we have a maximum flow?



Max flow = 11

Flow networks

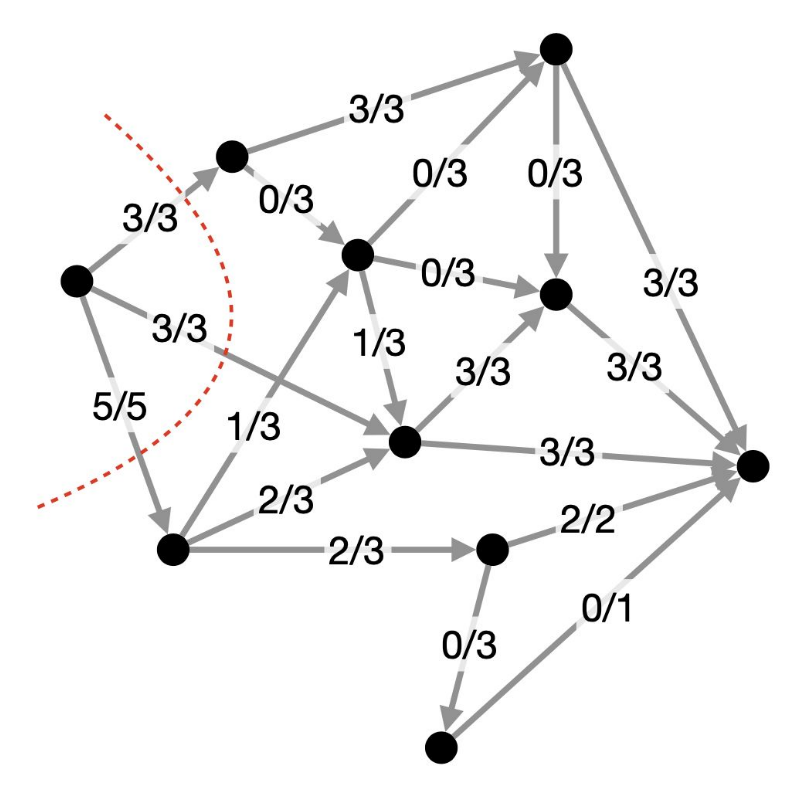
- What is a flow network?
- What is max flow?
- How to increase flow?
- How do you know if it is possible to increase flow?
- How to prove we have a maximum flow?
  - the residual network has no more augmenting paths



Max flow = 11

Flow networks

- What is a flow network?
- What is max flow?
- How to increase flow?
- How do you know if it is possible to increase flow?
- How to prove we have a maximum flow?
- What is the relation between maximum flow and minimum cut?



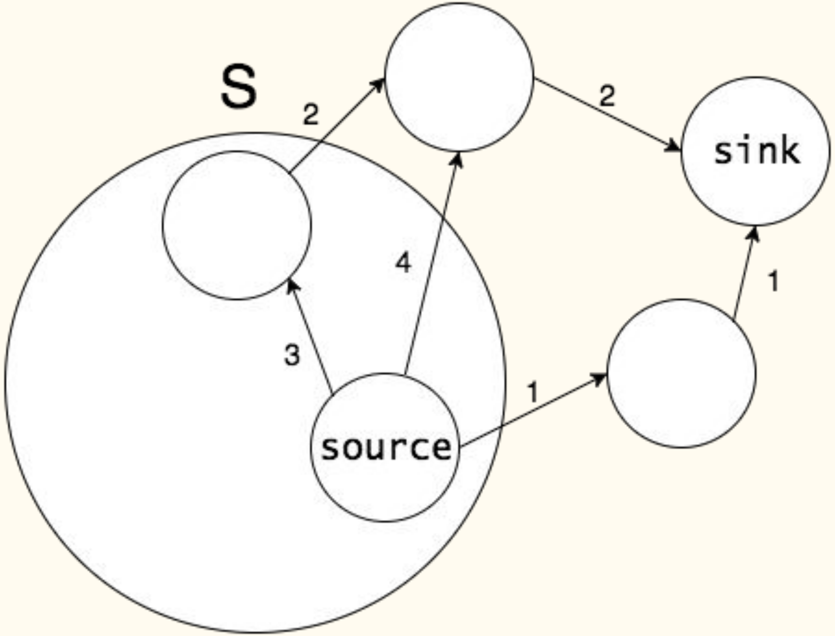
Max flow = 11

Flow networks

- A cut is a partitioning of the network G, into two disjoint sets of vertices.
- These sets are called S and T. S is the set that includes the source, and T is the set that includes the sink.
- The only rule is that the source and the sink cannot be in the same set.
- A cut has two important properties.

The first is the cut-set, which is the set of edges that start in S and end in T.

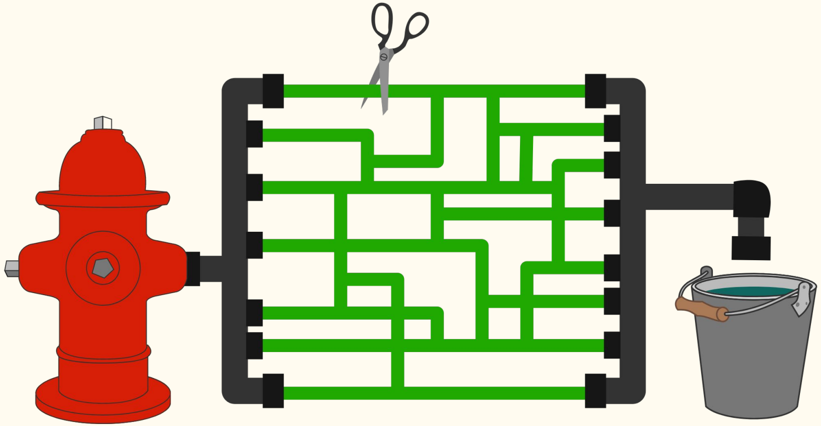
The second is the capacity, which is the sum of the weights of the edges in the cut-set.



Flow networks

The max-flow min-cut theorem states that:

For any network graph and a selected source and sink node, the max-flow from source to sink = the min-cut necessary to separate source from sink.

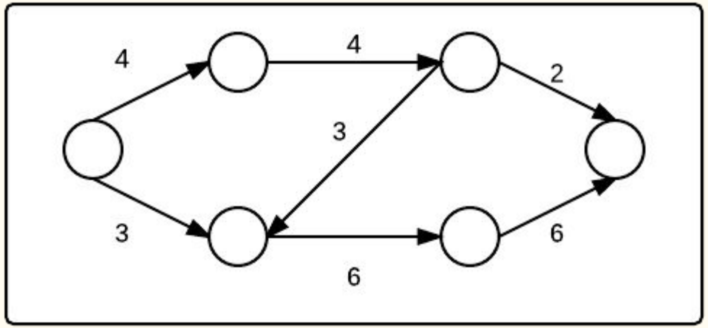


Practice time

https://bit.ly/dsa-2020-max-flow-min-cut

1. Go to the link to generate a random flow network
2. Find maximum flow

Ford-Fulkerson Algorithm



- The intuition goes like this: as long as there is a path from the source to the sink that can take some flow the entire way, we send it. This path is called an augmenting path.
- We keep doing this until there are no more augmenting paths.
- In the image above, we could start by sending 2 cars along the topmost path (because only 2 cars can get through the last portion).
- Then we might send 3 cars along the bottom path for a total of 5 cars.
- Finally, we can send 2 more cars along the top path for two edges, send them down to bottom path and through to the sink.
- The total number of cars sent is now 7, and it is the maximum flow.

Ford-Fulkerson Algorithm

```
initialize flow to 0
path = findAugmentingPath(G, s, t)
while path exists:
    augment flow along path
    G_f = createResidualGraph()
    path = findAugmentingPath(G_f, s, t)
return flow
```

Ford-Fulkerson Algorithm

```
flow = 0
for each edge (u, v) in G:
    flow(u, v) = 0
while there is a path, p, from s -> t in residual network G_f:
    residual_capacity(p) = min(residual_capacity(u, v) : for (u, v) in p)
    flow = flow + residual_capacity(p)
    for each edge (u, v) in p:
        if (u, v) is a forward edge:
            flow(u, v) = flow(u, v) + residual_capacity(p)
        else:
            flow(u, v) = flow(u, v) - residual_capacity(p)
return flow
```

Ford-Fulkerson Algorithm

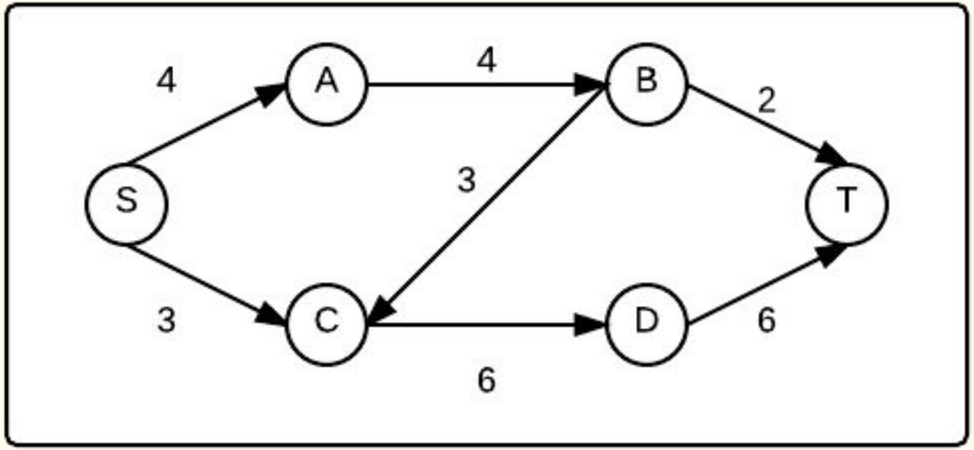
- When a residual graph, *G_f* is created, edges can be created that go in the opposite direction when compared to the original graph.
- An edge is a 'forward edge' if the edge existed in the original graph, *G*. If it is a reversal of an original edge, it is called a 'backwards edge.'
- Residual capacity is defined as the new capacity after a given flow has been taken away.

$$c_f(u, v) = c(u, v) - f(u, v).$$

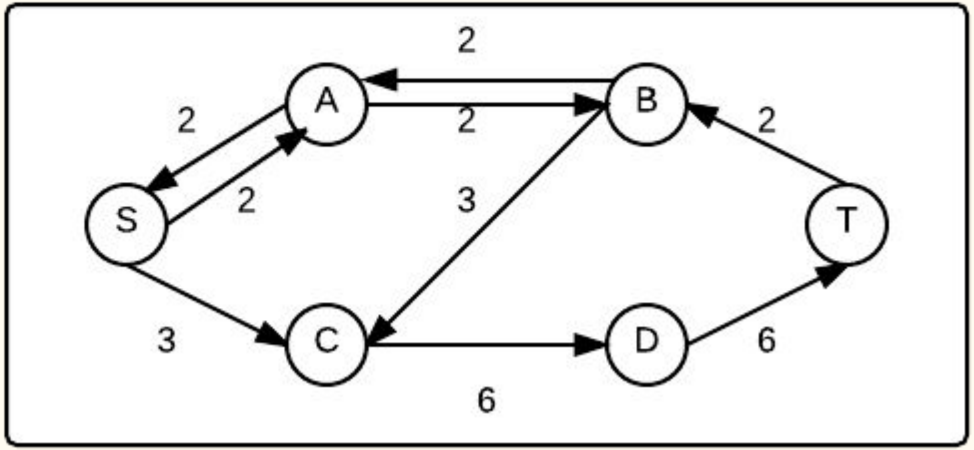
- However, there must also be a residual capacity for the reverse edge as well.

$$c_f(v, u) = c(v, u) + f(u, v).$$

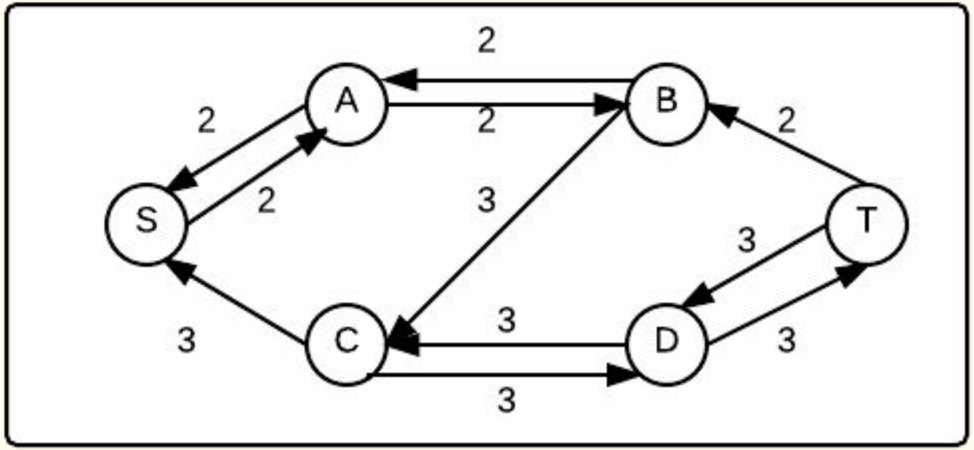
Ford-Fulkerson Algorithm



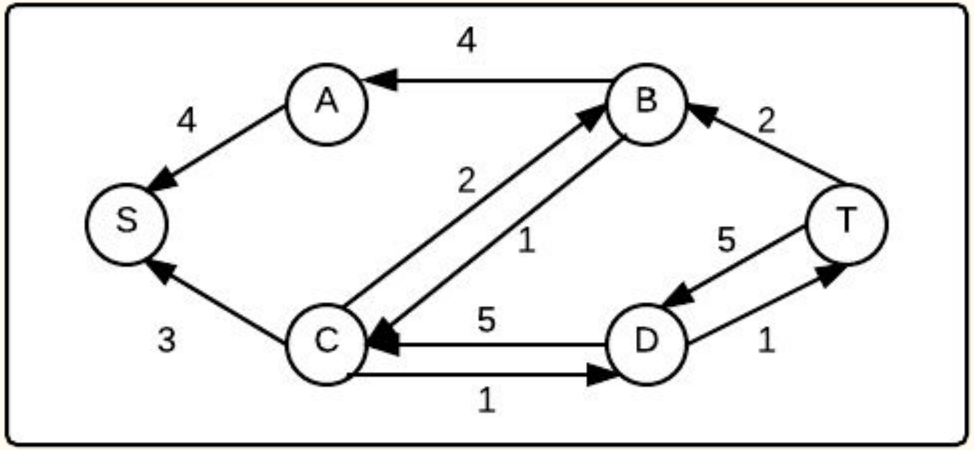
Ford-Fulkerson Algorithm



Ford-Fulkerson Algorithm



Ford-Fulkerson Algorithm



Live Coding
Ford-Fulkerson Algorithm