CSE 373
Data Structures & Algorithms

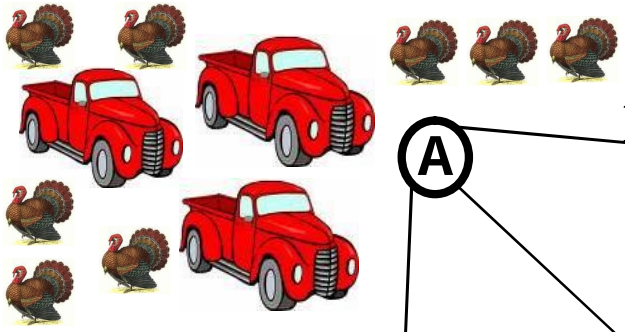
Lectures 23

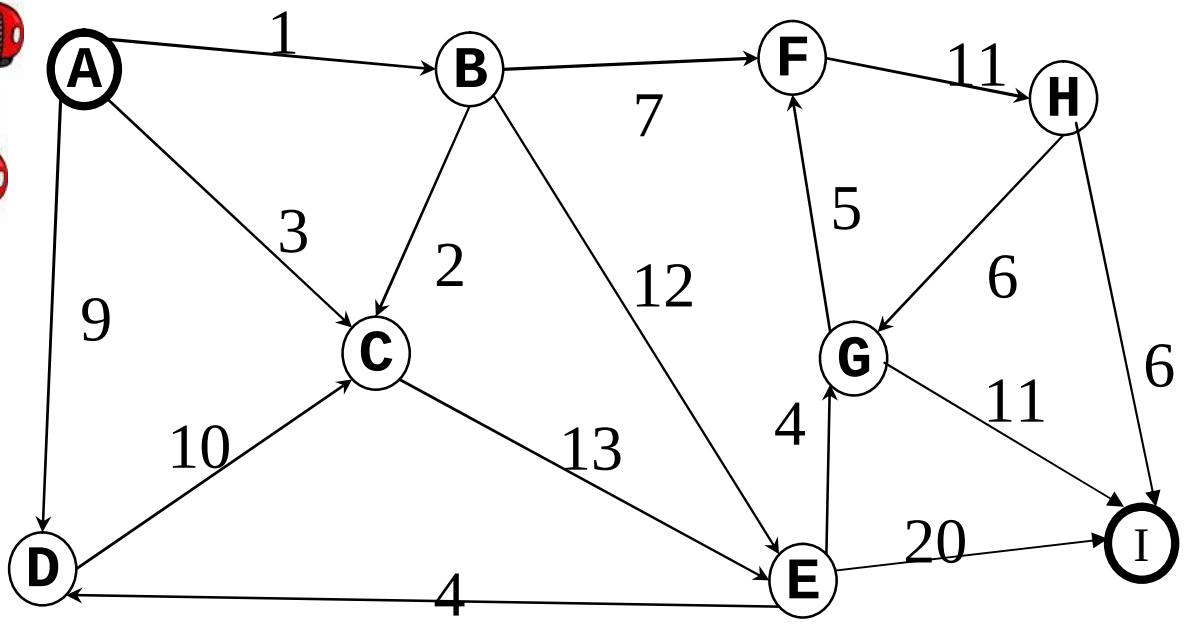
Network Flow

Network Flow

- Given a weighted, directed graph G=(V,E)
- Treat the edge weights as *capacities*
- How much can we flow through the graph?

Network Flow





Network Flow: Definitions

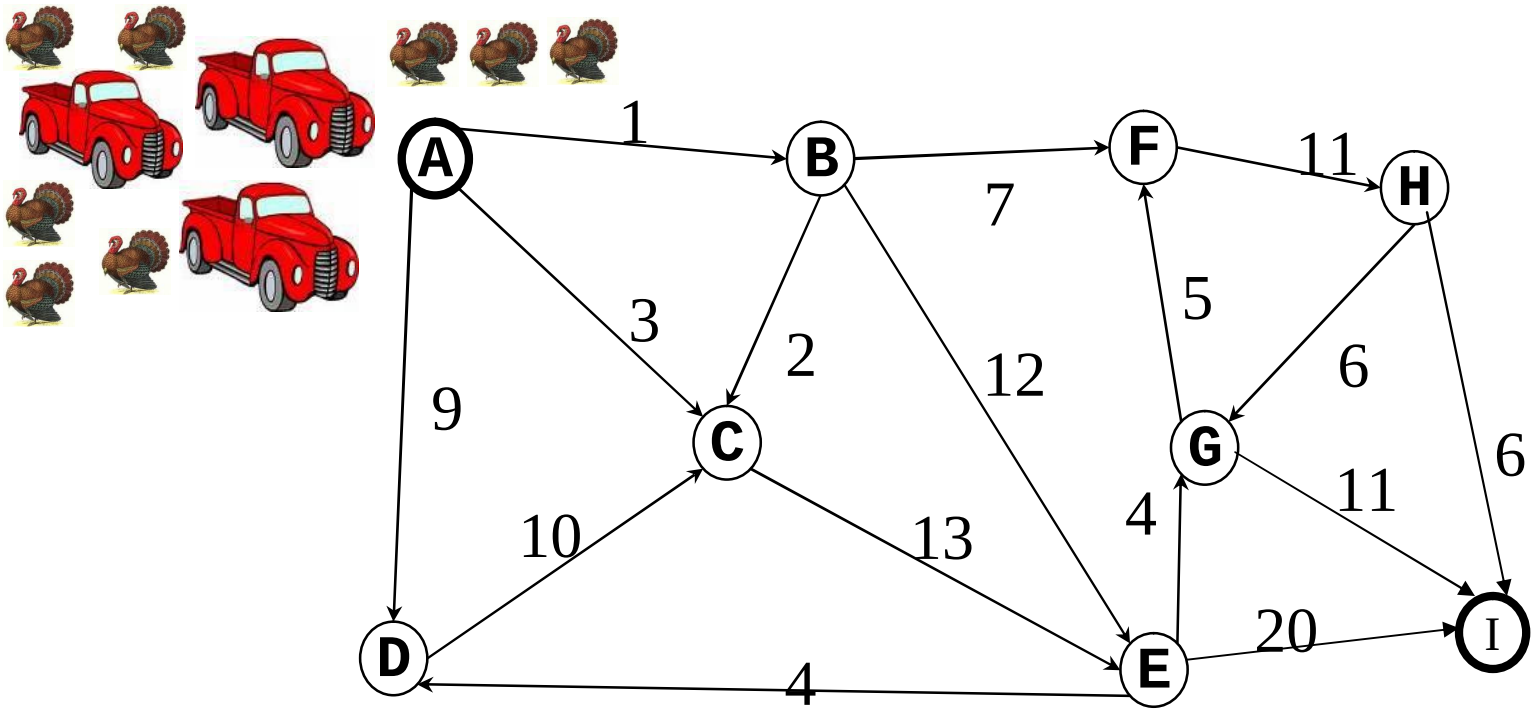
- Define two special vertices
  - *source* s
  - *sink* t
- Define a *flow* as a function on edges:
  - **Capacity**: $f(v,w) <= c(v,w)$
  - **Conservation**: $\sum_{v \in V} f(u,v) = 0$ for all u except source, sink
  - **Value of a flow**: $|f| = \sum_{v} f(s,v)$

Network flow: Definitions

- **Capacity:** We cannot overload an edge
- **Conservation:** Flow entering any vertex must equal flow leaving that vertex
- We want to maximize the **value of a flow**, subject to these constraints
- A **saturated edge** is at maximum capacity

Network Flow

- So, how do we want to go about this?

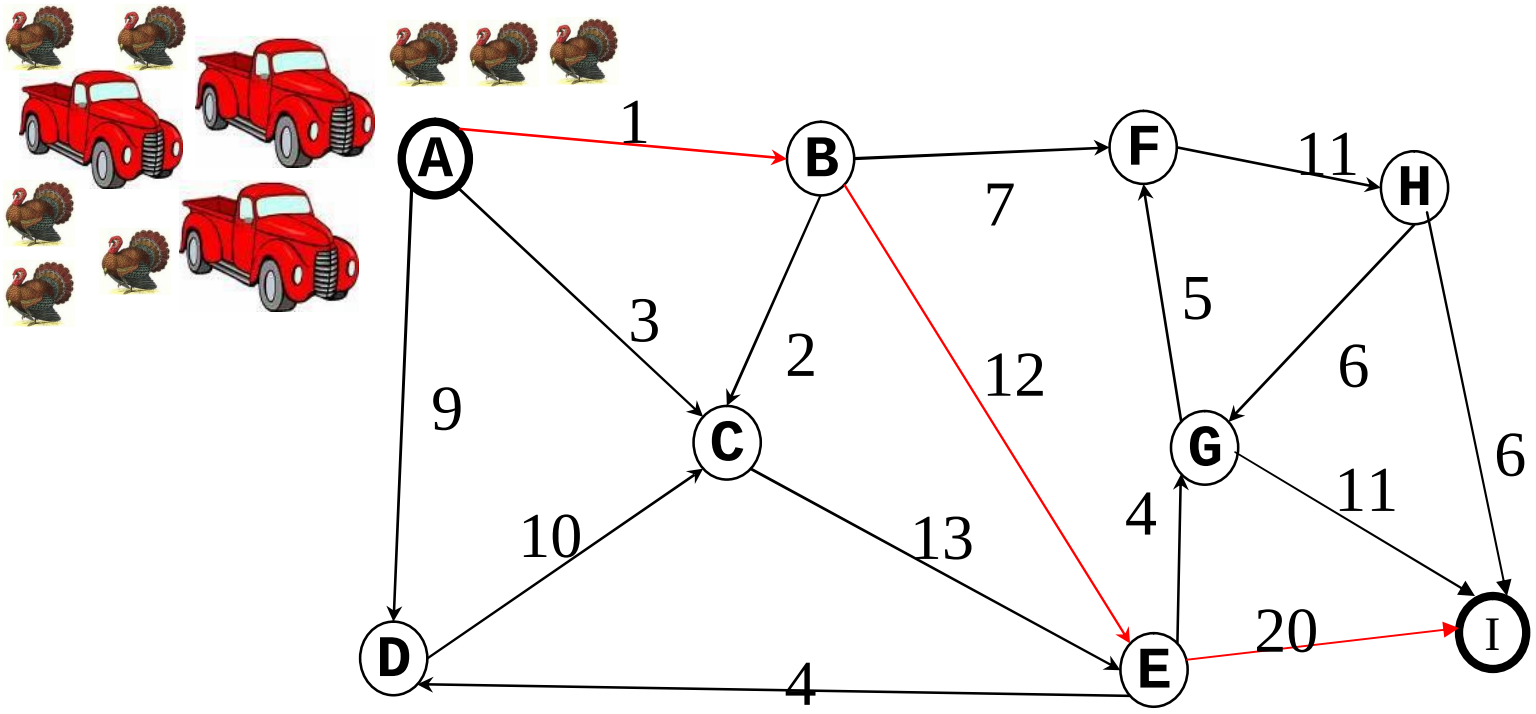


A Good Idea

- Start with flow 0
- "While there's room for more flow, push more flow across the network"
  - While there exists a path from *s* to *t*, none of whose edges are saturated
  - Push more flow along that path, until one of its edges is saturated
  - Known as finding an "augmenting path"

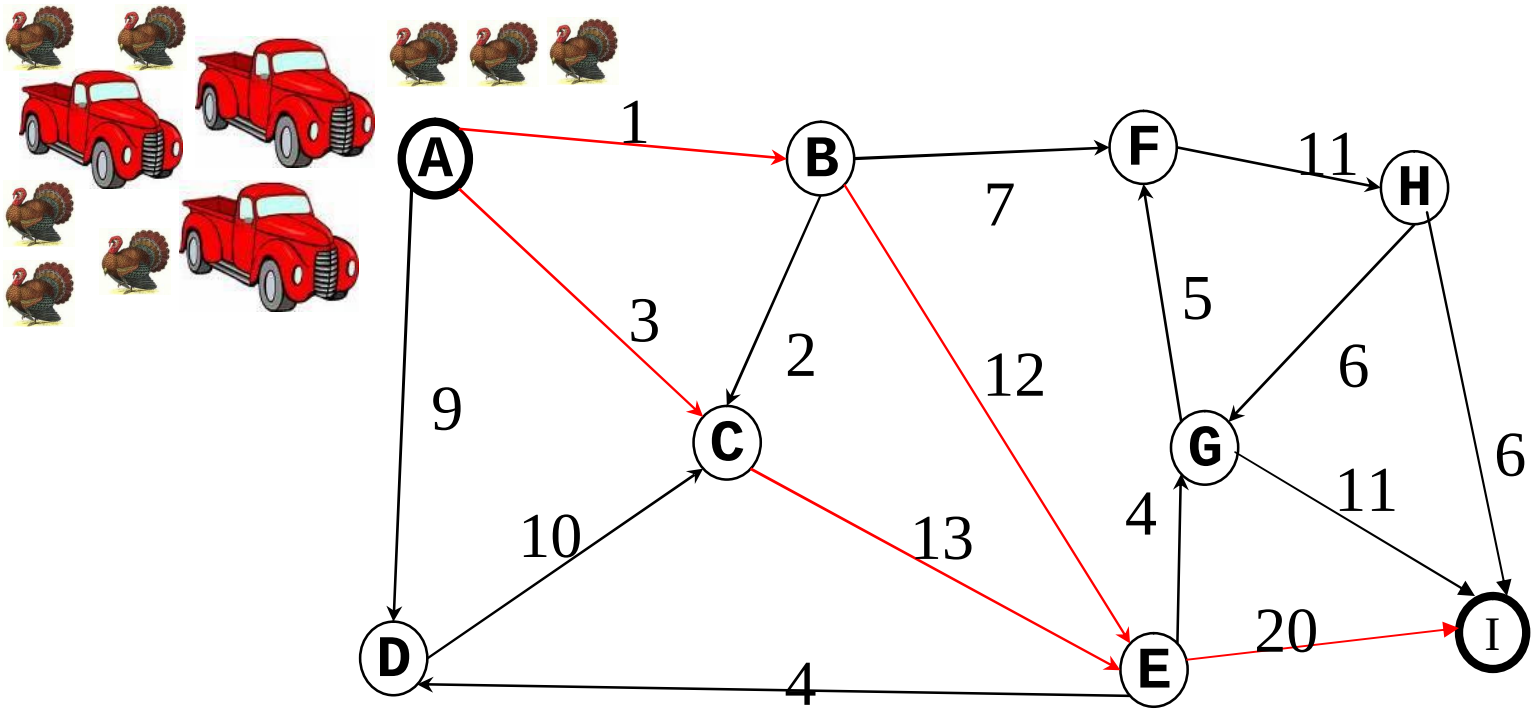
A Problem

- We should be able to use edges more than once, but how much do we have left?



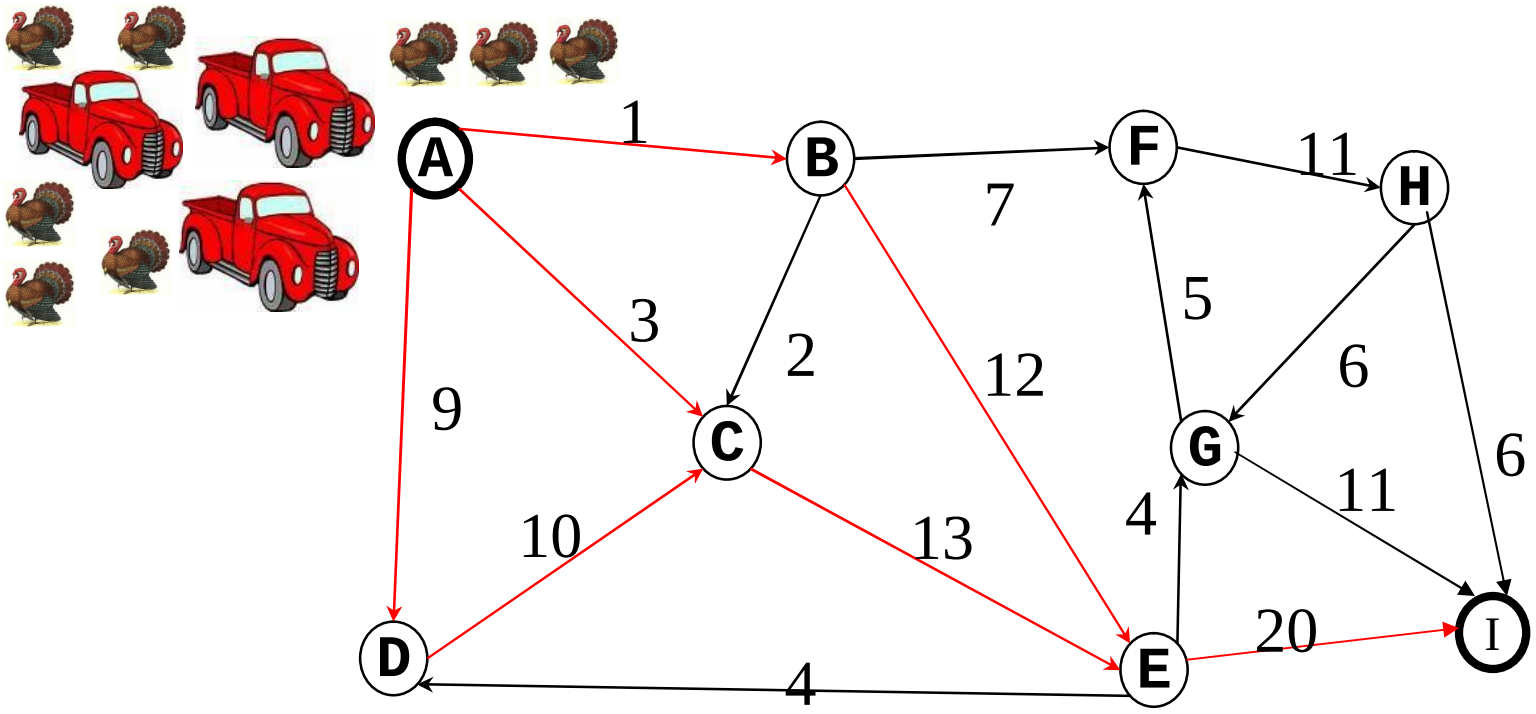
A Problem

- We should be able to use edges more than once, but how much do we have left?



A Problem

- We should be able to use edges more than once, but how much do we have left?

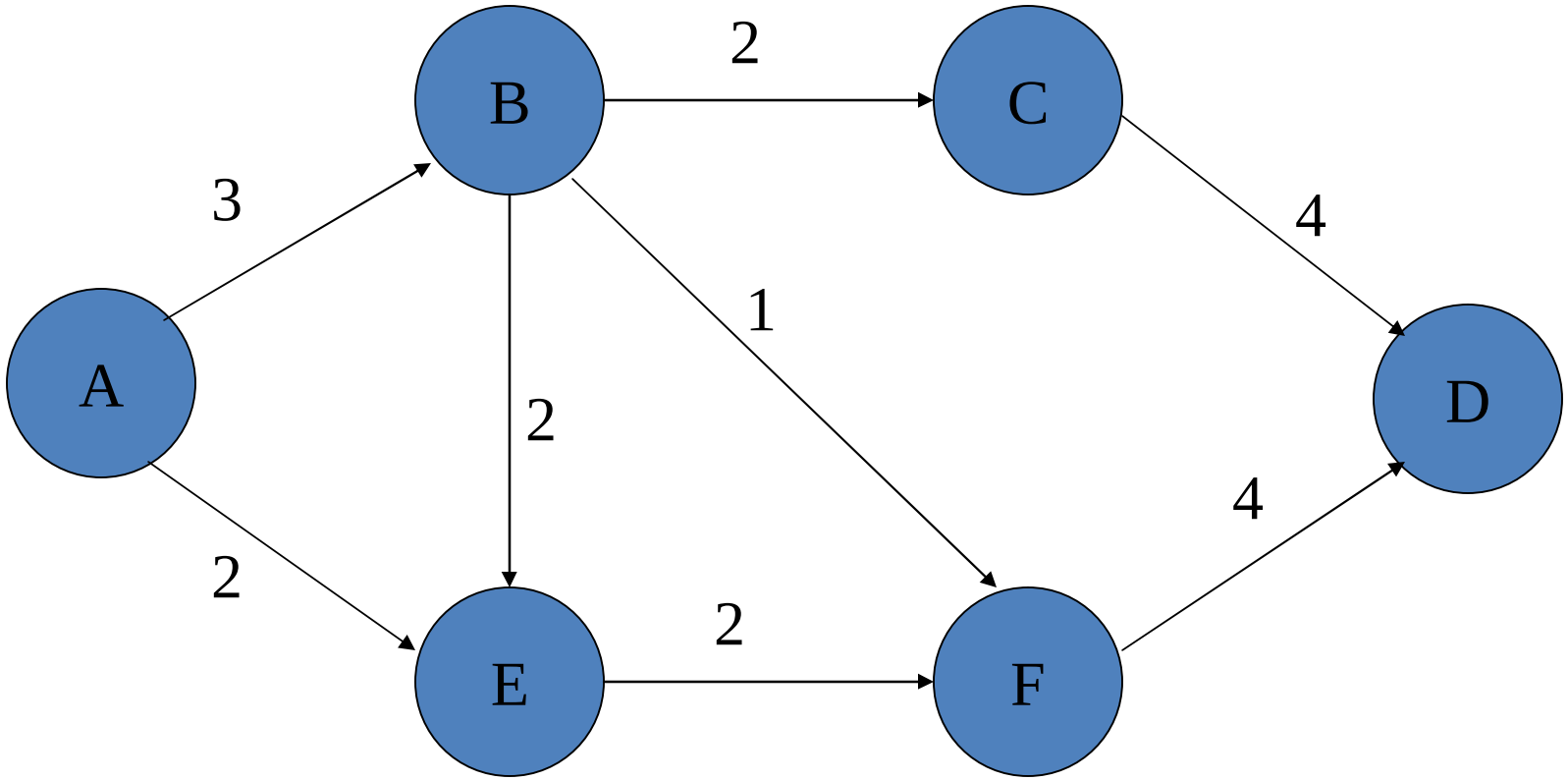


Residual Graph

- Constructing a residual graph:
  - Use the same vertices
  - Edge weights are the remaining capacity on the edges, given the existing augmenting paths
  - If there is a path from *s* to *t* in the residual graph, then there is available capacity there

Example

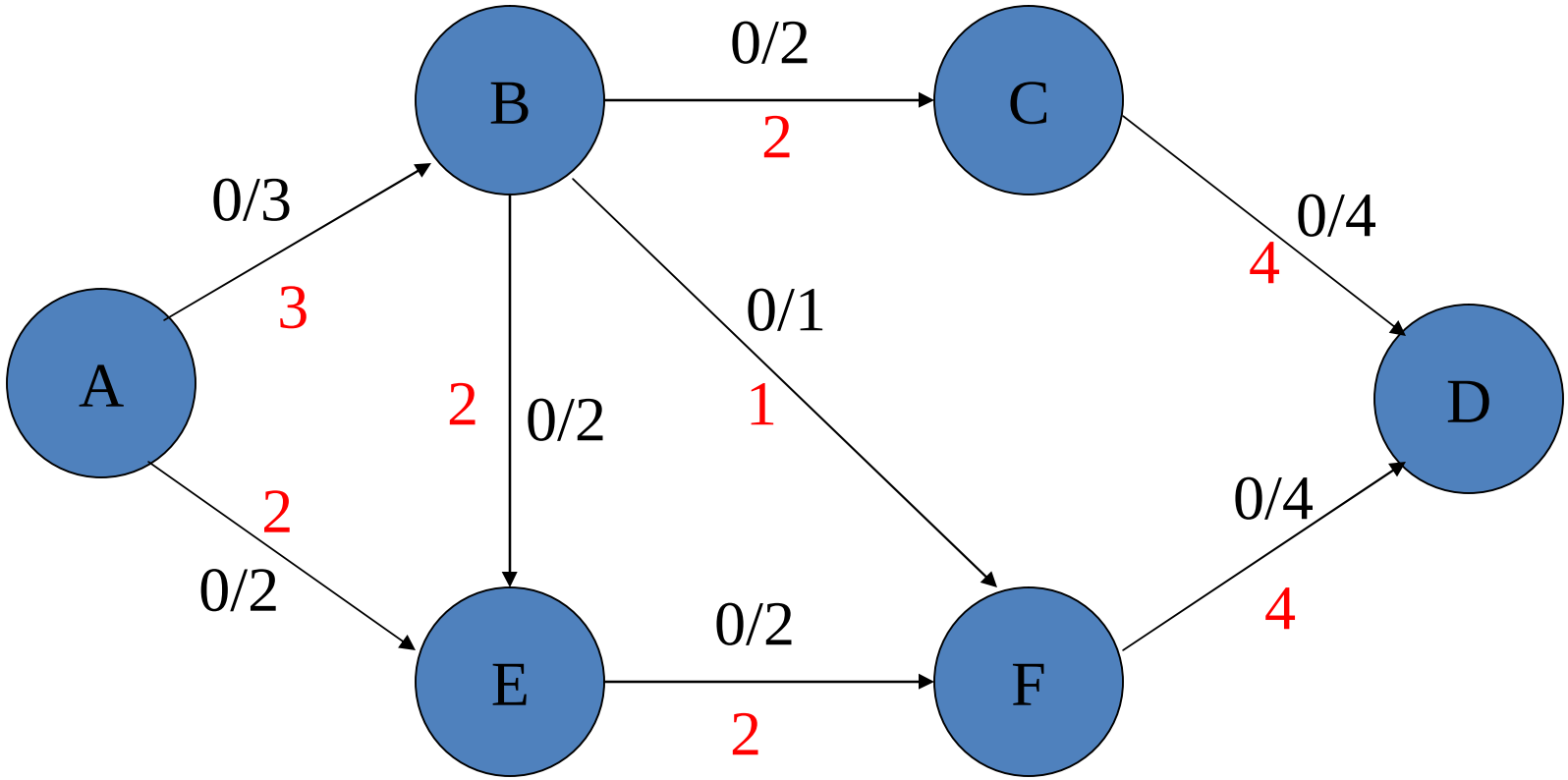
Initial Graph – No Flow



Capacity

Example

Include the residual capacities

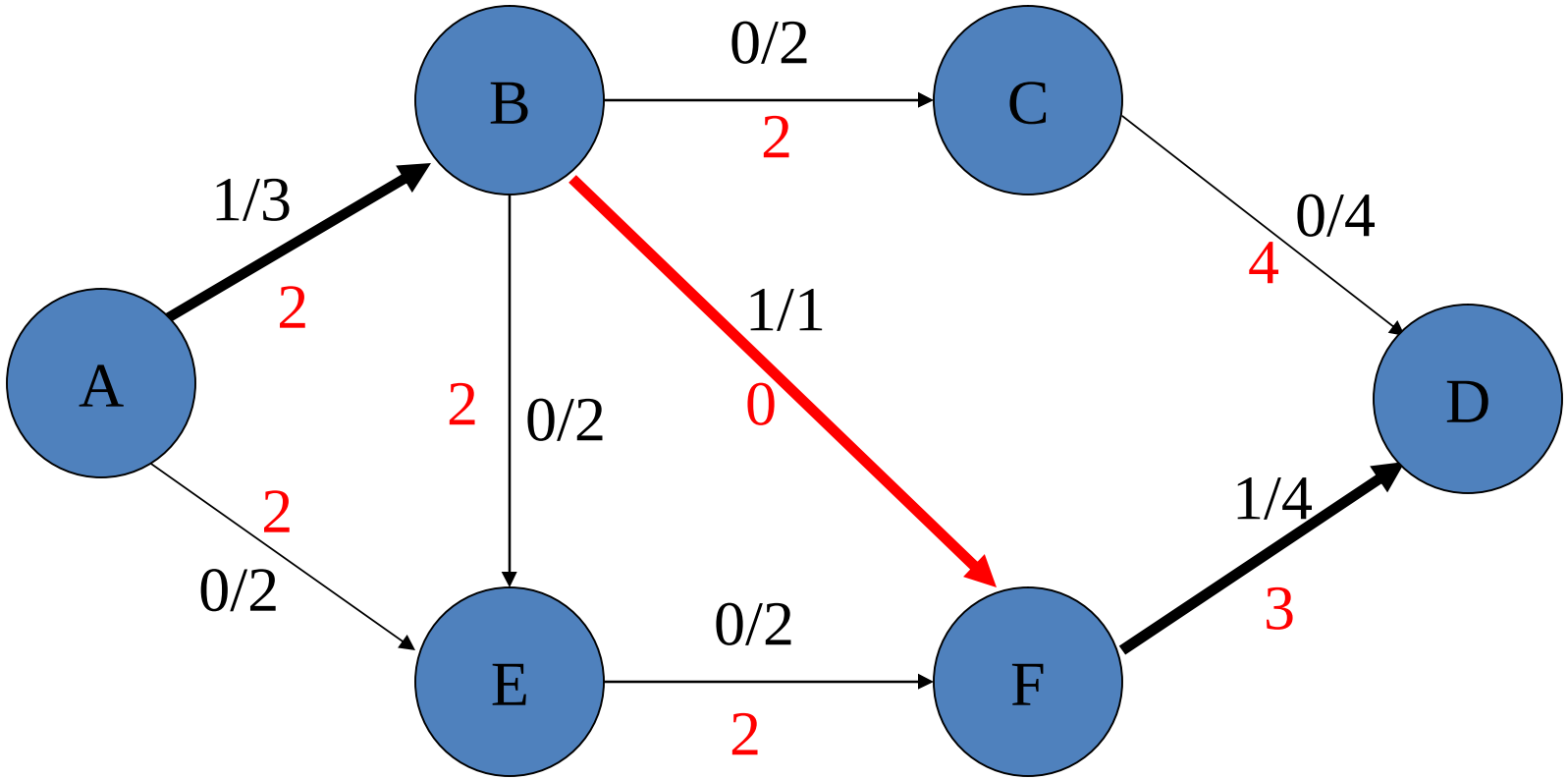


Flow / Capacity

Residual Capacity

Example

Augment along ABFD by 1 unit (which saturates edge BF)

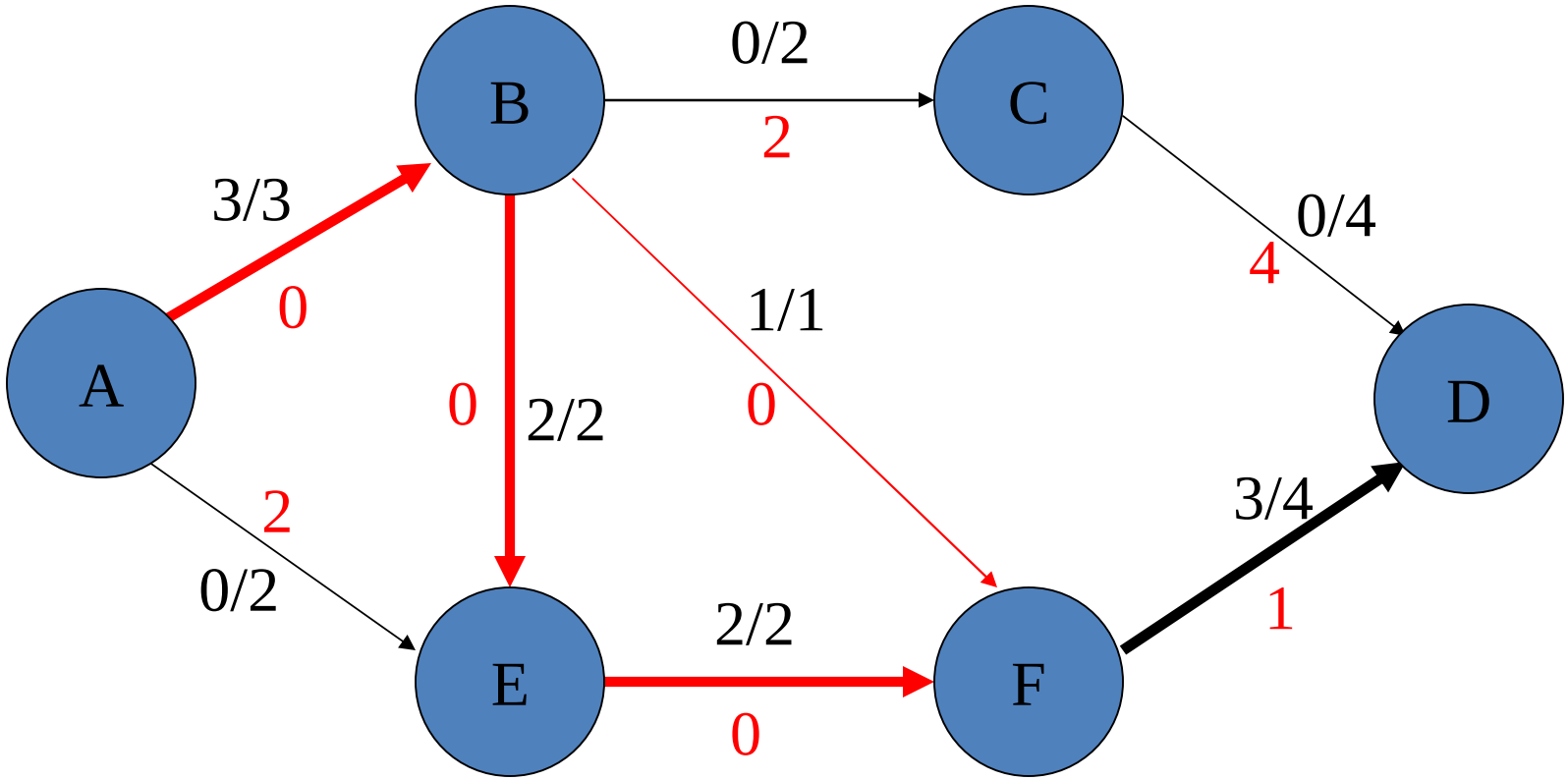


Flow / Capacity

Residual Capacity

Example

Augment along ABEFD by 2 units (which saturates BE and EF)

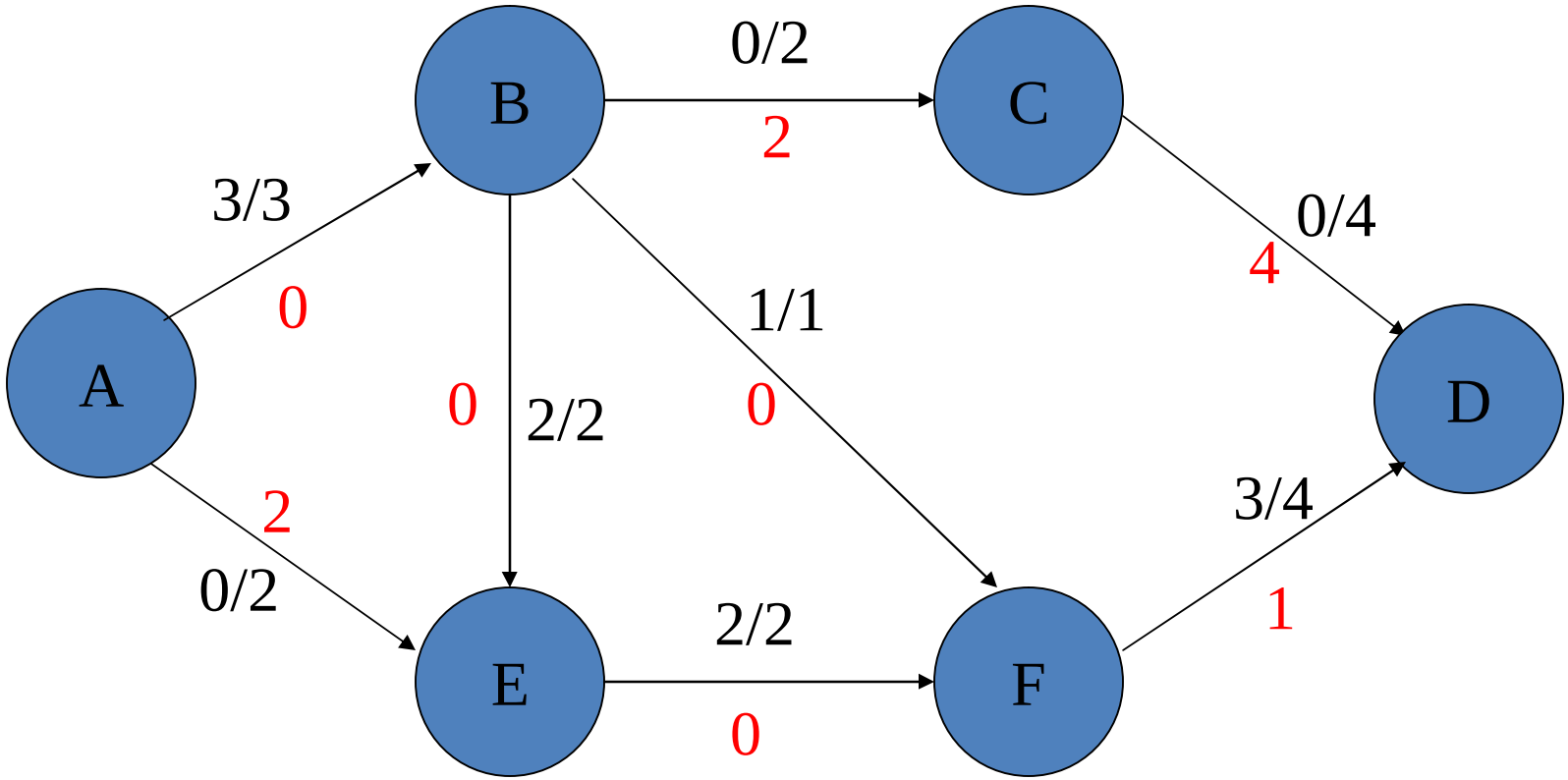


Flow / Capacity

Residual Capacity

Example

There are no paths in the residual graph, but we could fit more flow



Flow / Capacity

Residual Capacity

Ford-Fulkerson Method

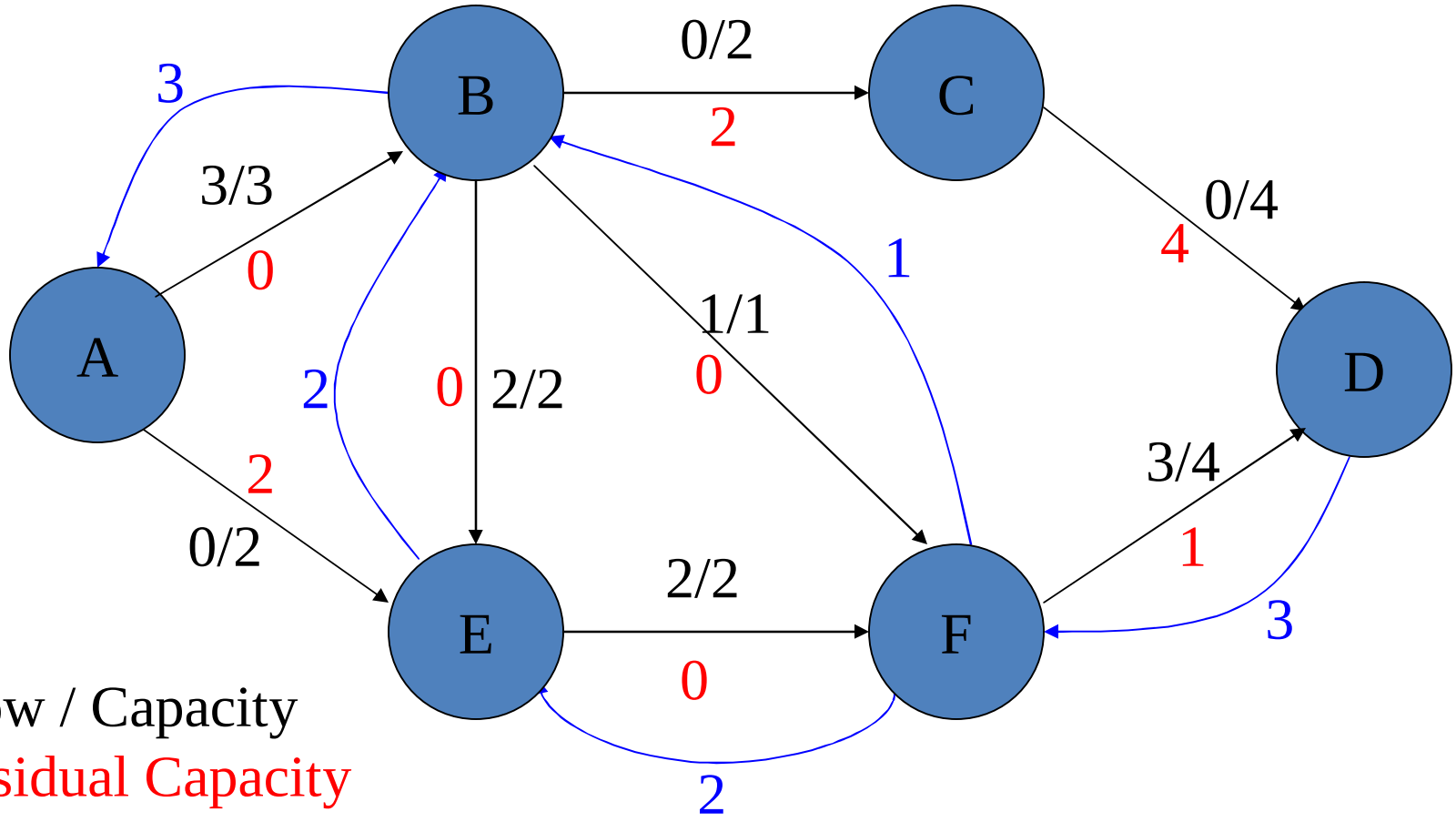
- Our greedy algorithm makes choices about how to route flow, and we never reconsider those choices
- Can we develop a way to efficiently reconsider the choices we already made?
- Can we do it by just modifying the graph?

Residual Graph

- Constructing a residual graph:
  - Use the same vertices
  - Edge weights are the remaining capacity on the edges, given the existing augmenting paths
  - Add additional edges for backward capacity
  - If there is a path from *s* to *t* in the residual graph, then there is available capacity there

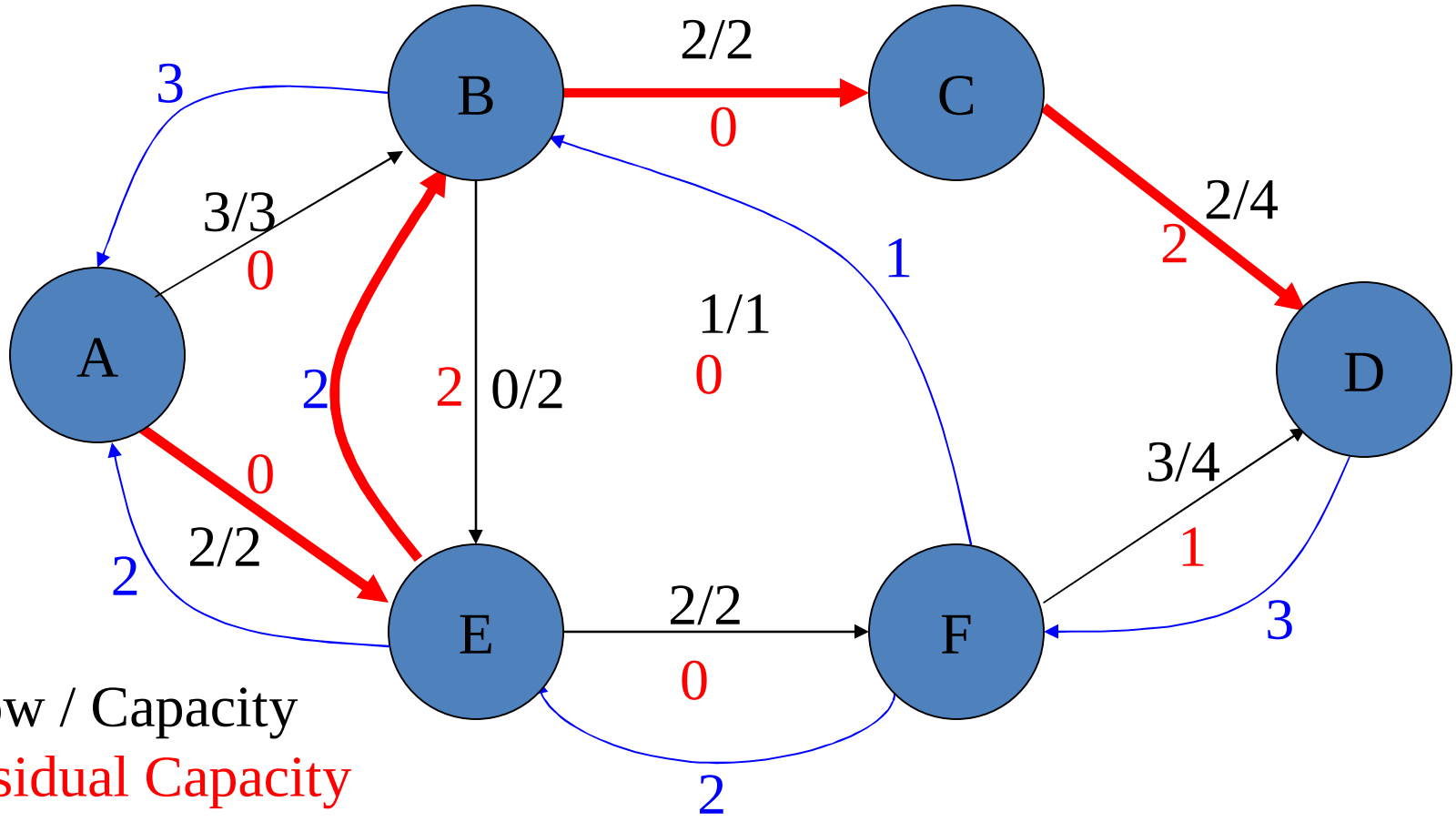
Example

Add the backwards edges, to show we can "undo" some flow



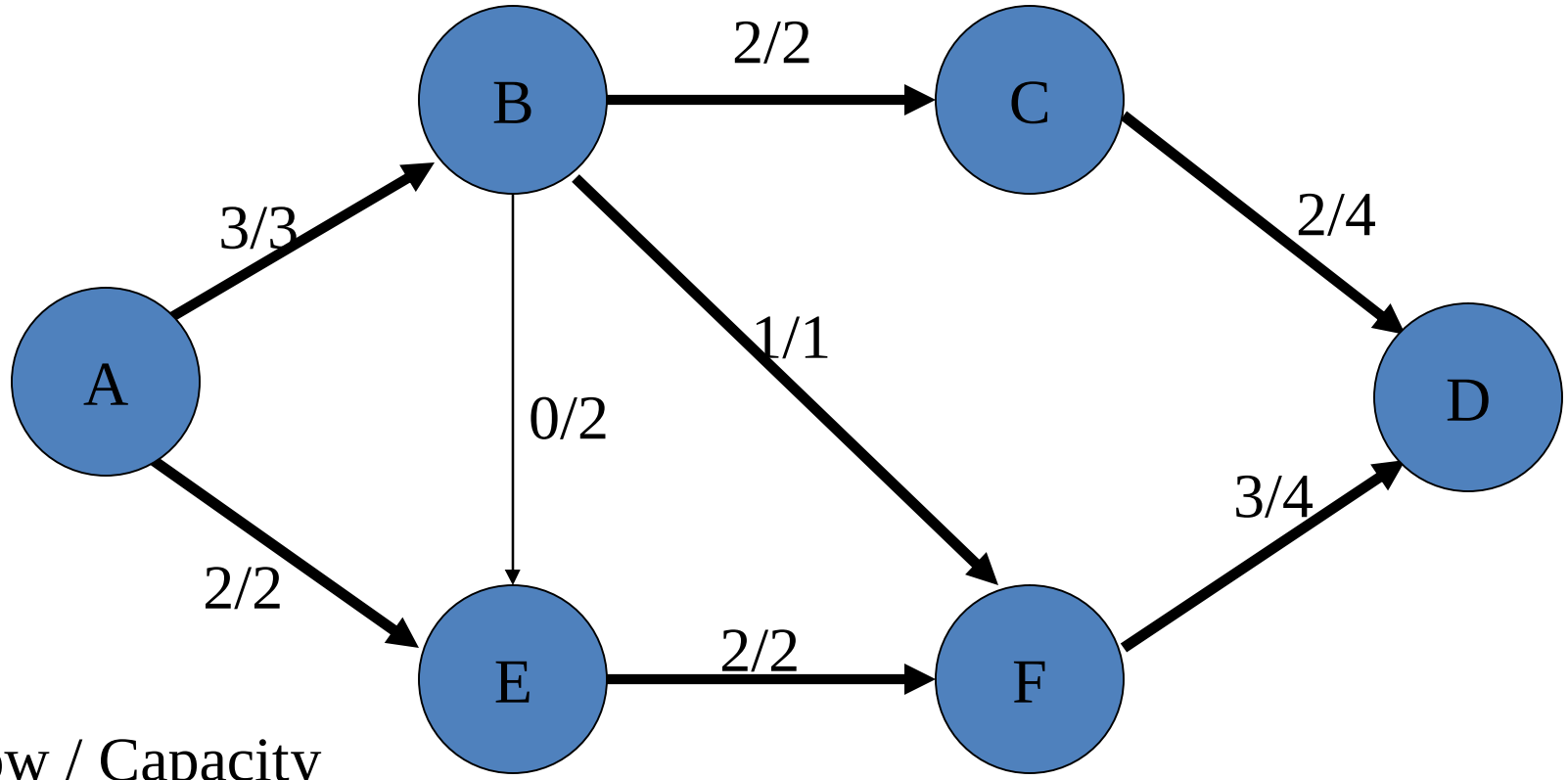
Example

Augment along AEBCD (which saturates AE and EB, and empties BE)



Example

Final, maximum flow



Flow / Capacity

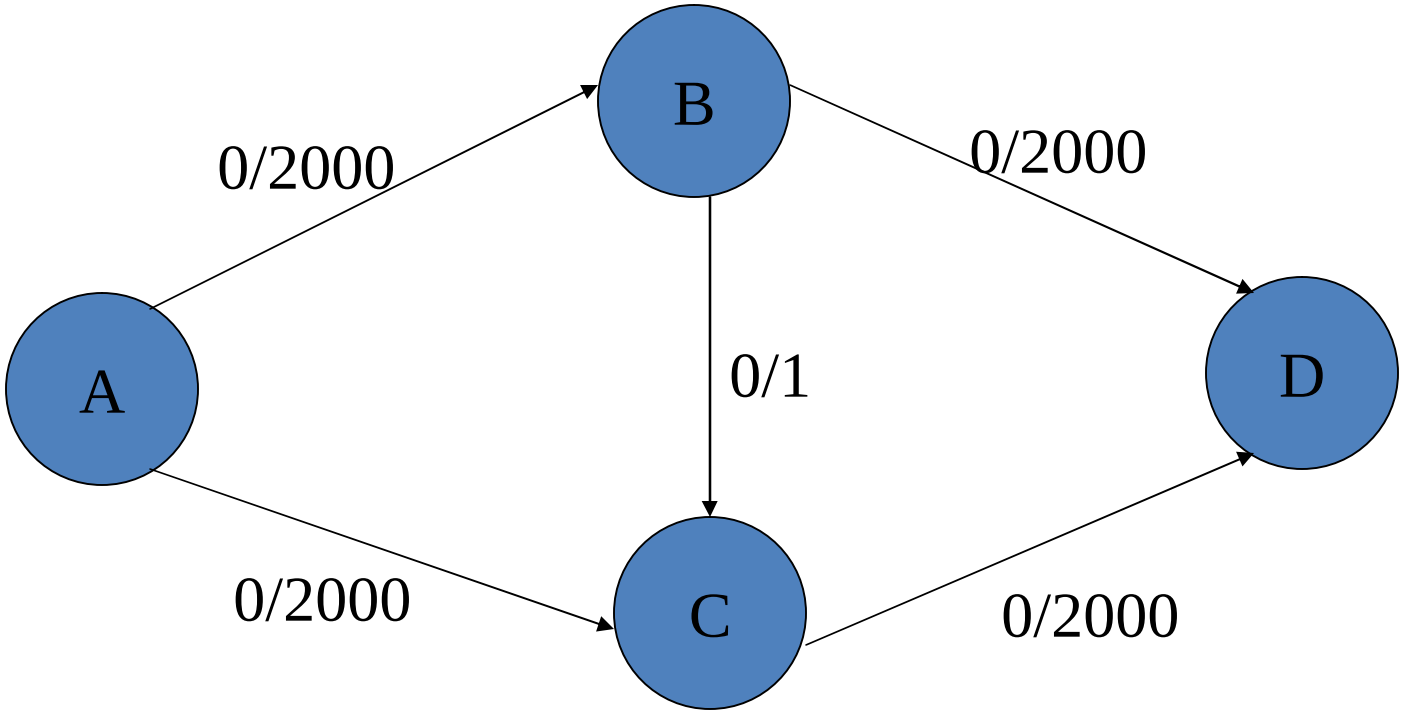
Residual Capacity

Backwards flow

How should we pick paths?

- Two very good heuristics (Edmonds-Karp):
  - Pick the largest-capacity path available
    - Otherwise, you'll just come back to it later…
  - Pick the shortest augmenting path available
    - For a good example why…

Don’t Mess this One Up



Augment along ABCD, then ACBD, then ABCD, then ACBD…

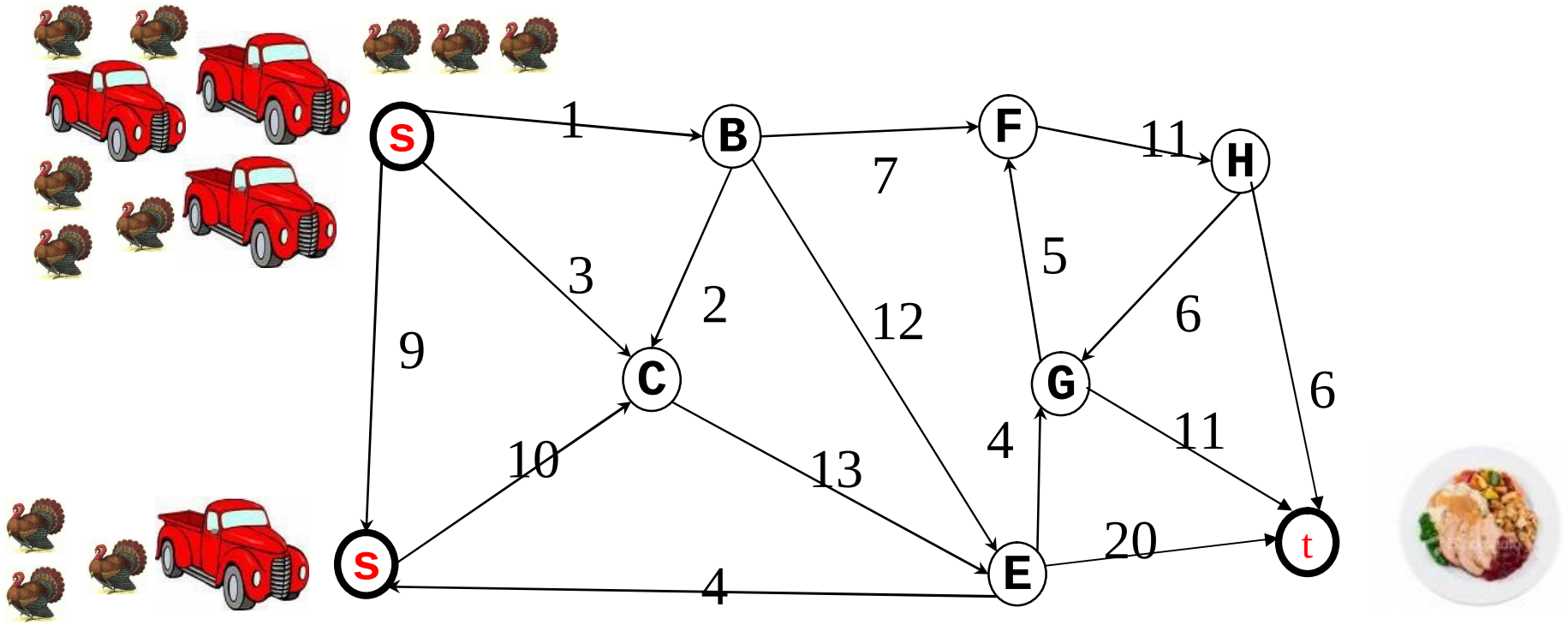
Should just augment along ACD, and ABD, and be finished

Running Time

- If always adding shortest path
- Each augmenting path can't be shorter, and it can't always stay the same length
  - So we have at most O(E) augmenting paths to compute for each possible length, and there are only O(V) possible lengths.
  - Each path takes O(E) time to compute
- Total time = $O(E^2V)$

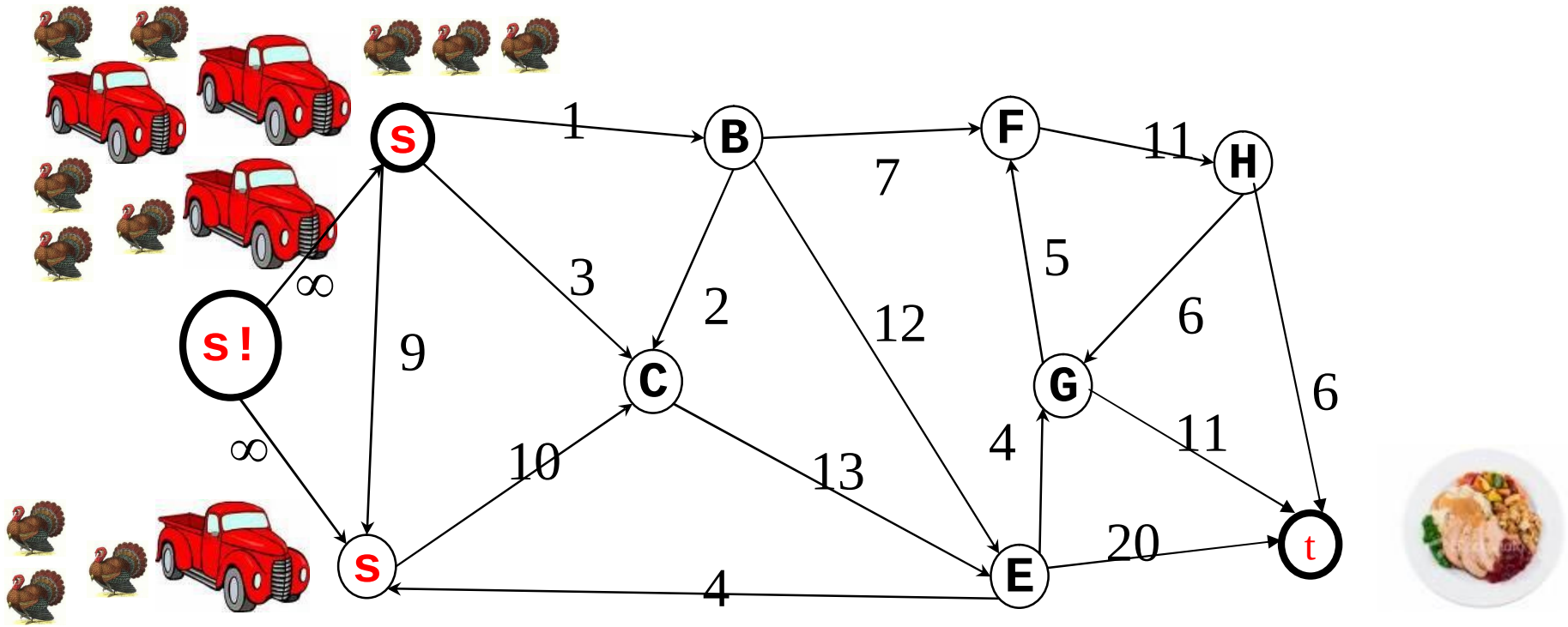
Network Flows

- What about multiple turkey farms?



Network Flows

- Create a single source, with infinite capacity edges connected to sources



Network Flows

- What if each farm only has a limited number of turkeys?
- What if the FDA will catch you if you route too many turkeys through particular some nodes?
- What if turkeys need to visit processing plants before they get to people?
- What if each plant can only handle a limited number of tourist turkeys?
- What if you need to make a profit?

One More Flow Definition

- We can talk about the flow from a set of vertices to another set, instead of just from one vertex to another:

$$f(X,Y) = \sum_{x \in X} \sum_{y \in Y} f(x,y)$$

- The only thing that counts is flow between the two disjoint sets of vertices

Network Cuts

- Intuitively, a cut separates a graph into two disconnected pieces

- Formally, a cut is a pair of sets (S, T):

$$V = S \cup T$$

$$S \cap T = \{\}$$

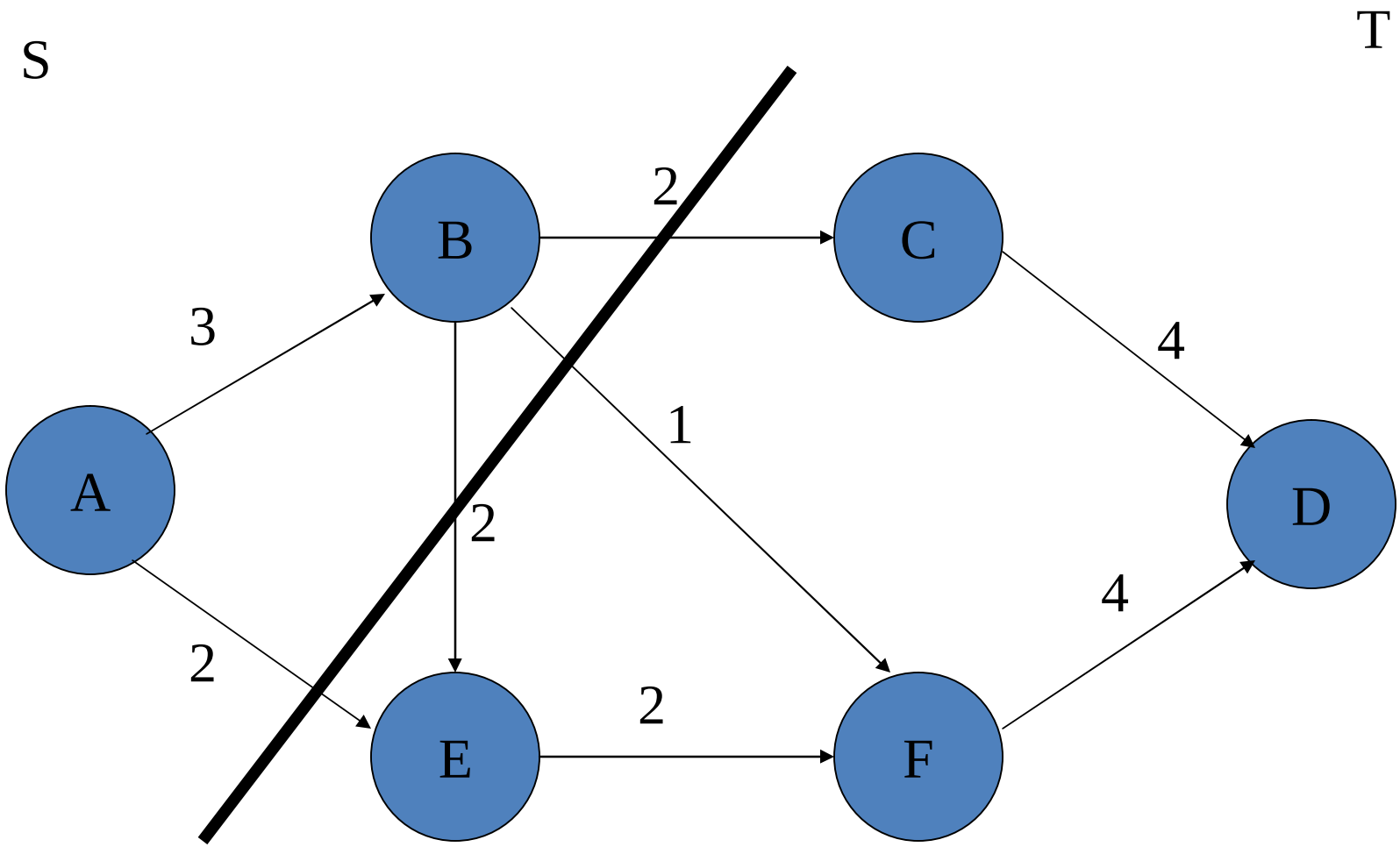
and S and T are connected subgraphs of G

Minimum Cuts

- If we cut G into (S, T), where S contains the source s and T contains the sink t,

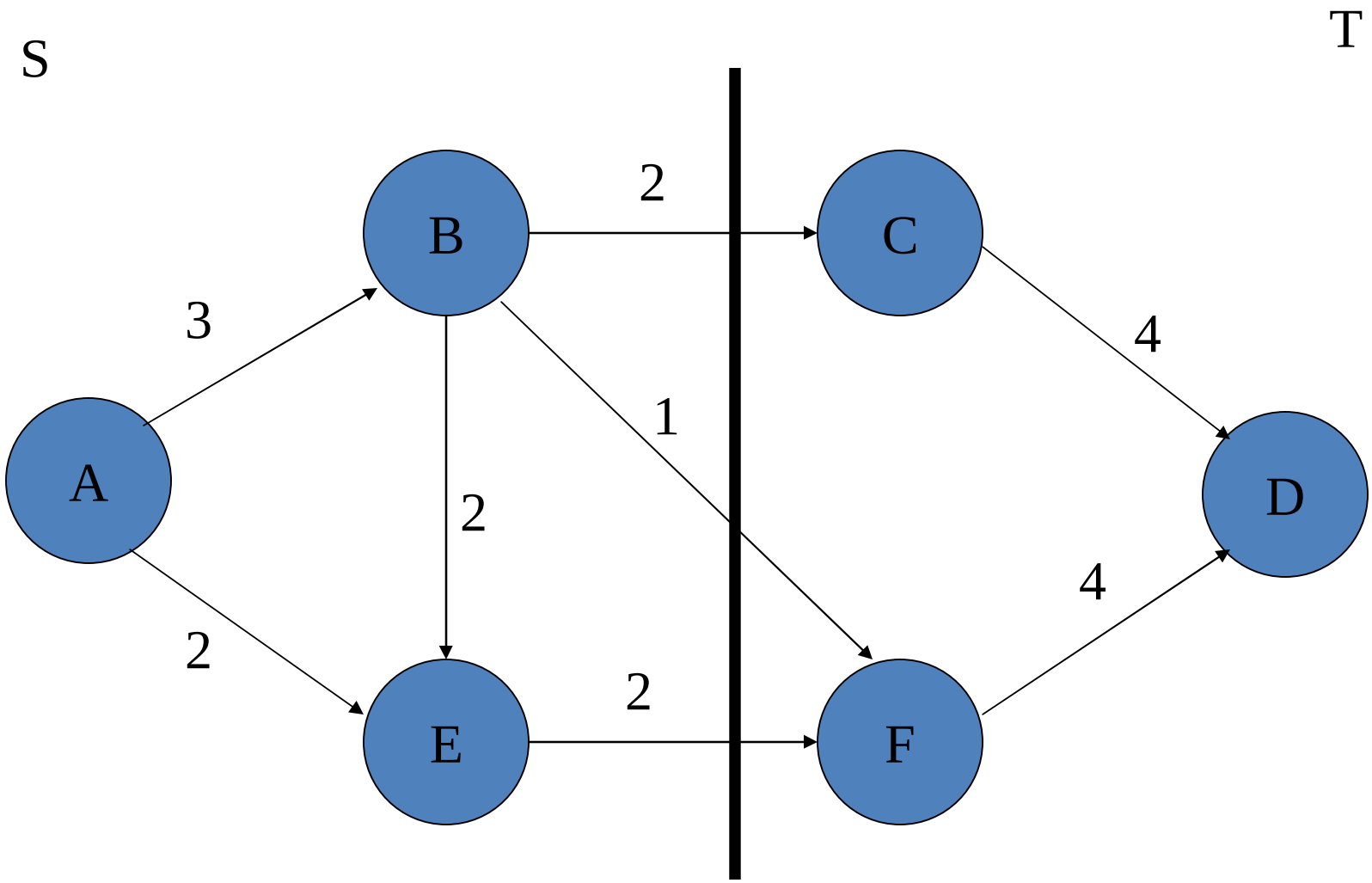
- Of all the cuts (S, T) we could find, what is the smallest (max) flow f(S, T) we will find?

Cut - Example



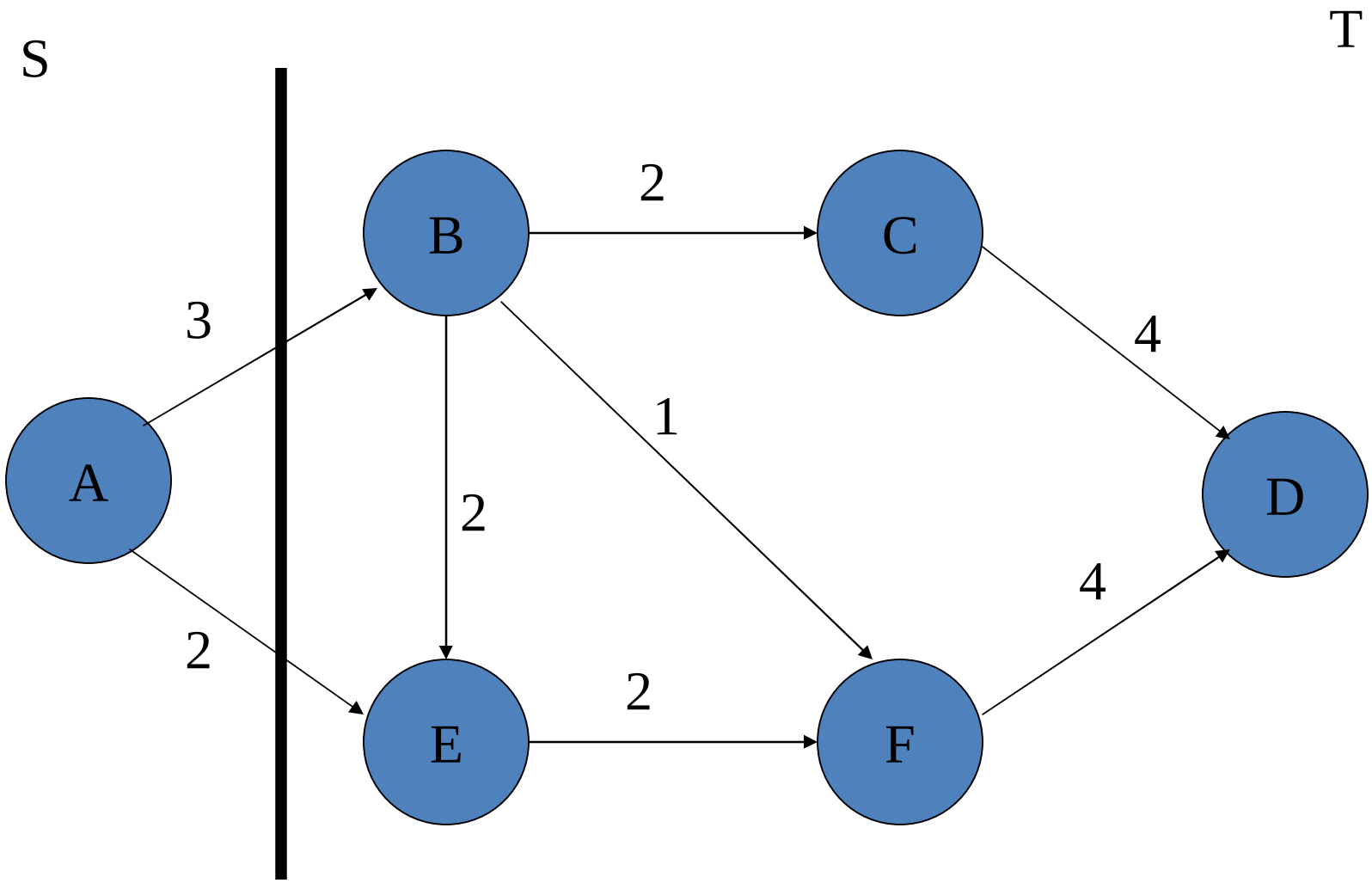
Capacity of cut = 7

Min Cut - Example



Capacity of cut = 5

Min Cut - Example



Capacity of cut = 5

Coincidence?

- No, Max-flow always equals Min-cut
  - If there is a cut with capacity equal to the flow, we have a maxflow:
    - We can't have a flow that's bigger than the capacity cutting the graph! So any cut puts a bound on the maxflow, and if we have an equality, then we must have a maximum flow.
  - If we have a maxflow, then there are no augmenting paths left
    - Or else we could augment the flow along that path, which would yield a higher total flow.
  - If there are no augmenting paths, we have a cut of capacity equal to the maxflow
    - Pick a cut (S,T) where S contains all vertices reachable in the residual graph from s, and T is everything else. Then every edge from S to T must be saturated (or else there would be a path in the residual graph). So c(S,T) = f(S,T) = f(s,t) = |f| and we're done.