**Max-flow, min-cut theorem**

The max-flow, min-cut theorem states that the maximum flow from a given source to a given sink in a network is equal to the minimum sum of a cut that separates the source from the sink.

**Overview**

The network discussed here is a directed graph G = (V, E) of vertices connected by edges with weights. Data flows from a source node (In-degree 0) to a sink node (Out-degree 0). Each weight denotes the capacity of the edge that represents the maximum flow through that edge. The theorem connects two quantities, maximum flow and minimum capacity cut. To further elaborate, we need to define the concept of flow and a cut.

**Flow**

For simplicity, flow can be imagined as a physical flow of a fluid through the network that moves from source to sink through the directed edges. Each edge will have a flow that cannot be greater than the edge’s capacity. Think of it as water flowing through a pipe, where the flow cannot be greater than the pipe’s capacity.

**Cut**

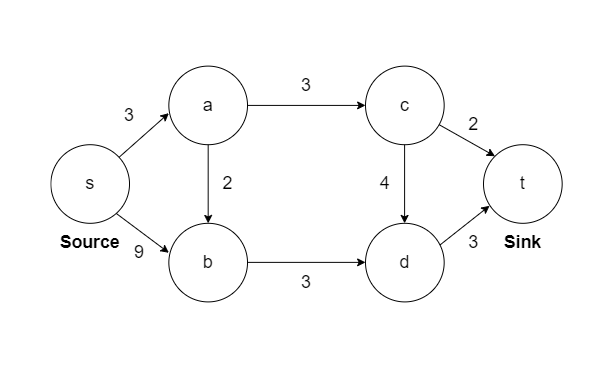
An s-t cut is partitioning the graph into two disjoint subsets, with the source in one part and sink in another. A cut will only be valid if it completely separates the source from the sink such as there is no path connecting the two. Alternatively, a cut is a set of edges whose removal divides the network into two halves X and Y, where s ∈ X and t ∈ Y. Moreover, a cut has a capacity that is equal to the sum of the capacities of the edges in the cut.

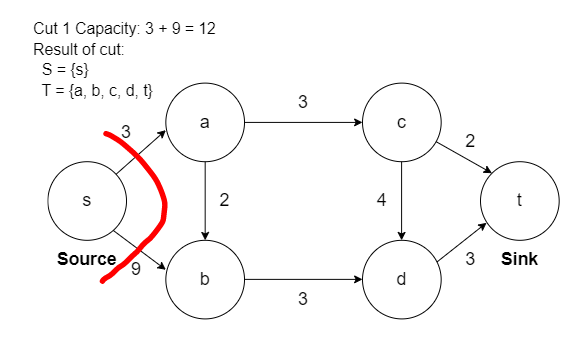
**Goal**

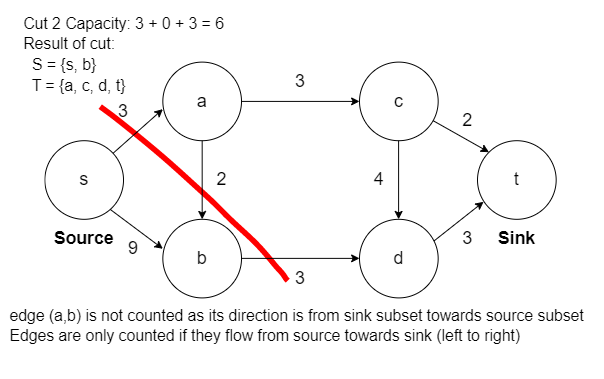
The goal is to find the minimum capacity cut that will dictate the maximum flow achievable in a flow network.

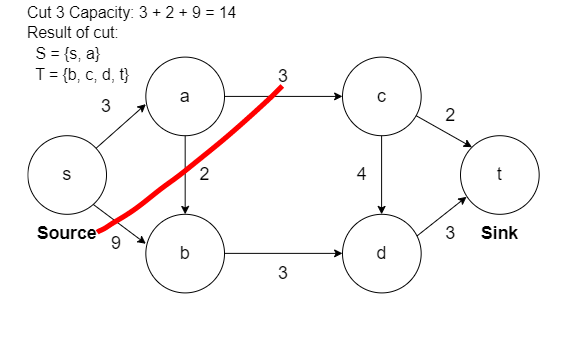
**Example**

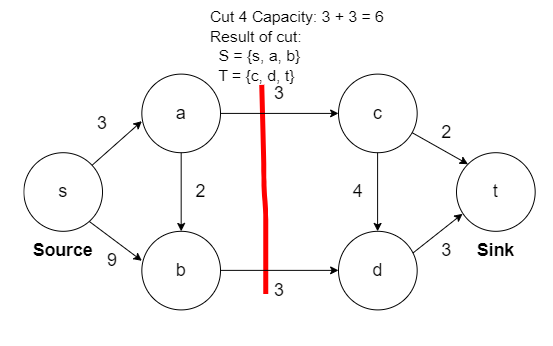
Let's look at an example of how to find a minimum cut in a network graph.

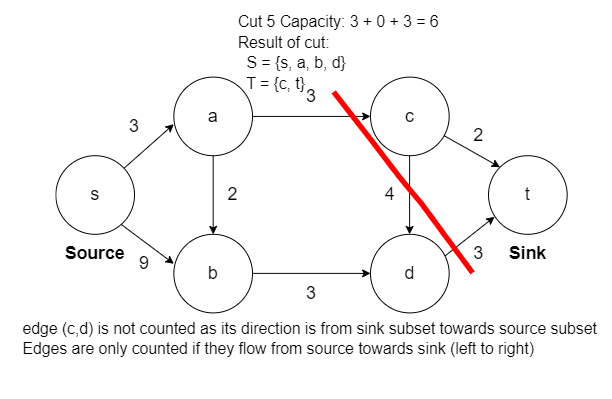


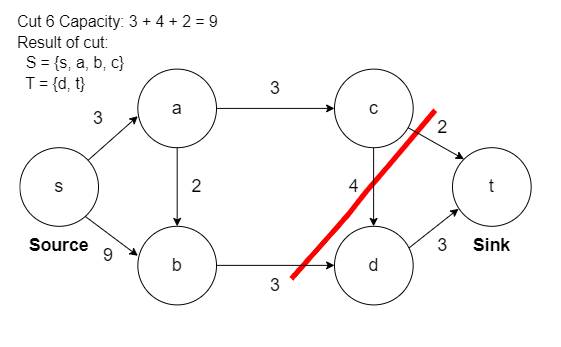


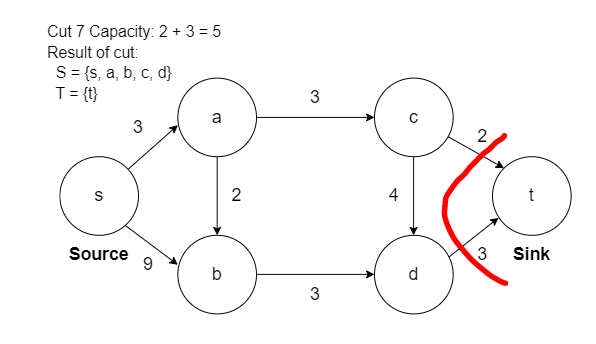


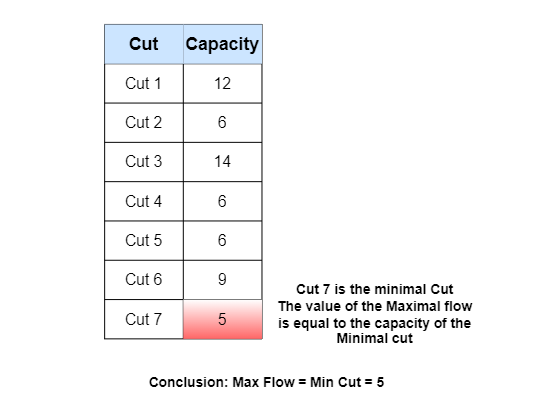












### Application

It is widely used in computer networks to maintain reliability and connectivity and used in Bipartite matching to match graphs.