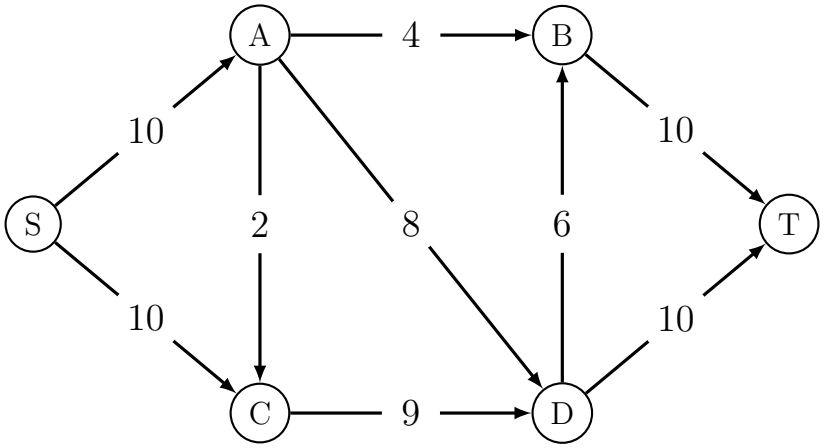
CSCI 270 - Spring 2023 - Discussion 8

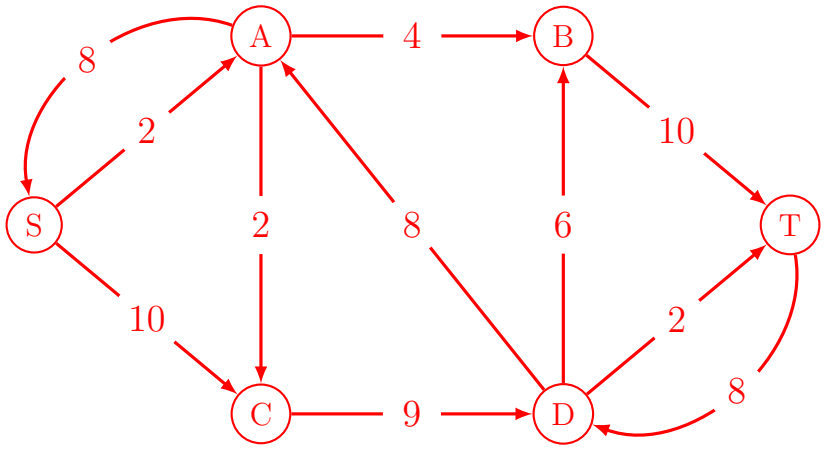
1. Consider the following graph with edge capacities.



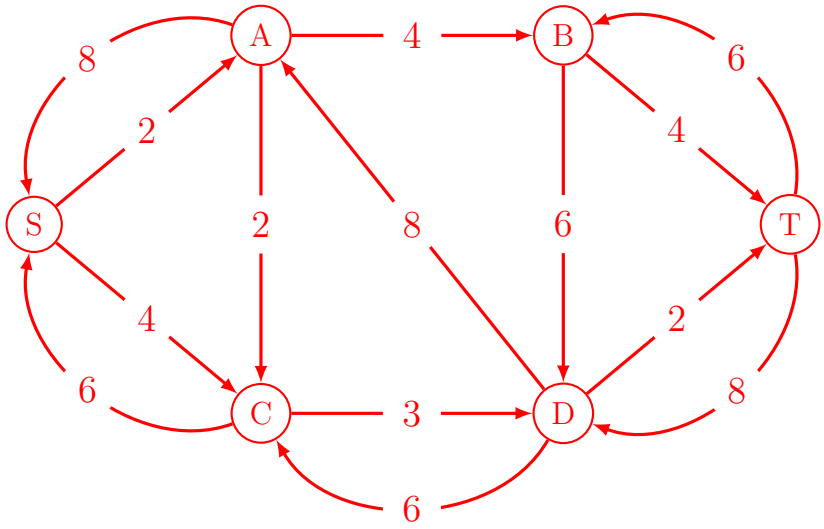
   (a) Run Ford-Fulkerson manually and confirm the residual graphs and the augmenting paths in each iteration.

The following is an example, but obviously, there are many different ways.

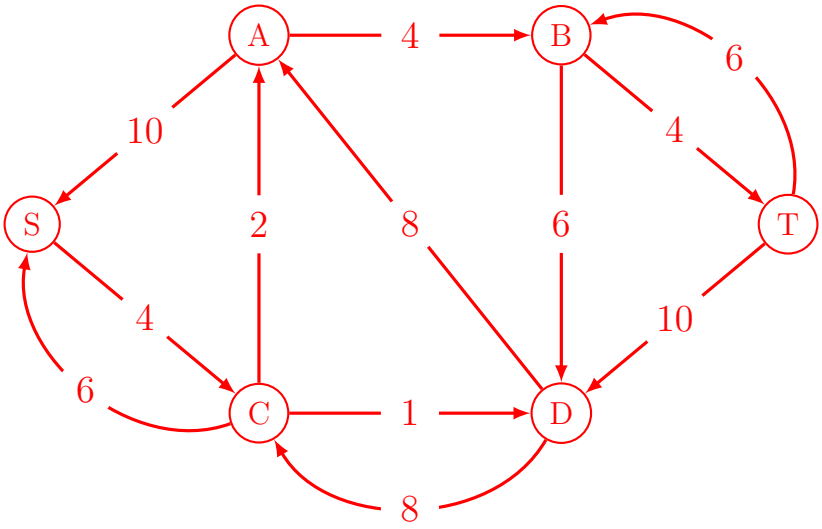
Push $S \to A \to D \to T$ with flow amount 8.



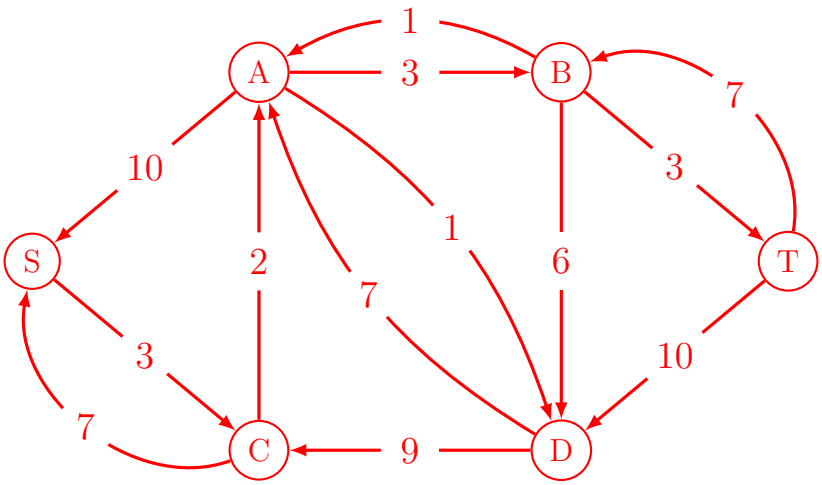
Push $S \to C \to D \to B \to T$ with flow amount 6.



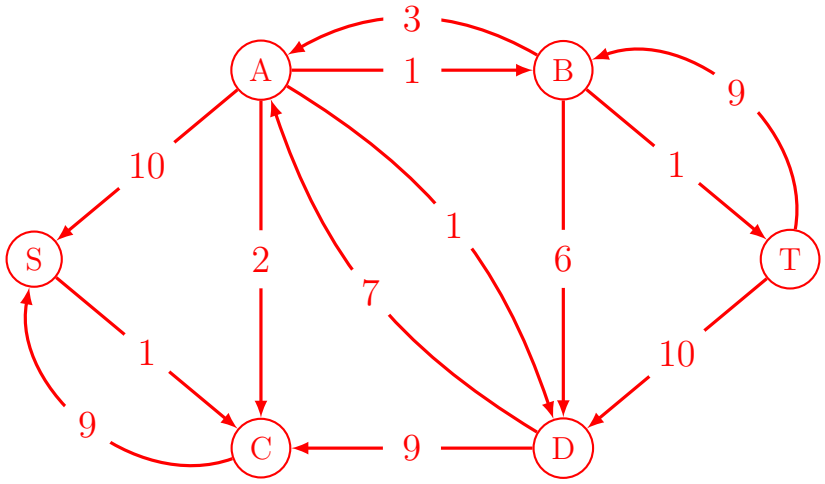
Push $S \to A \to C \to D \to T$ with flow amount 2.



Push $S \to C \to D \to A \to B \to T$ with flow amount 1.



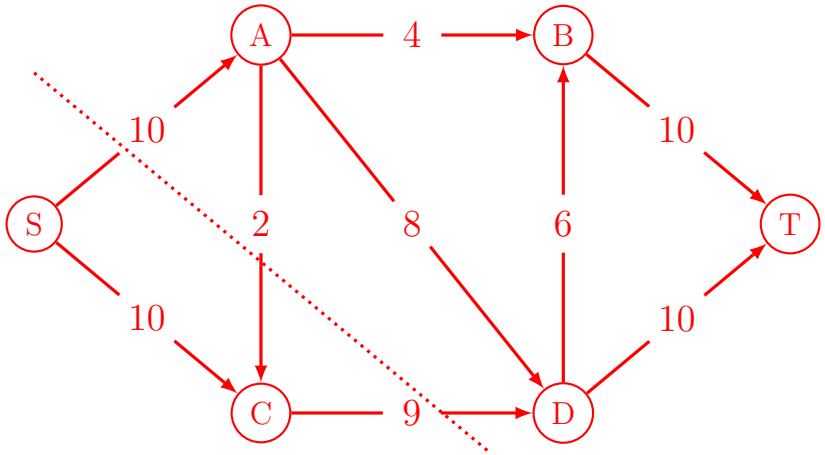
Push $S \to C \to A \to B \to T$ with flow amount 2.



The total amount of flow is 19.

(b) Give a min-cut and confirm min-cut = maxflow.

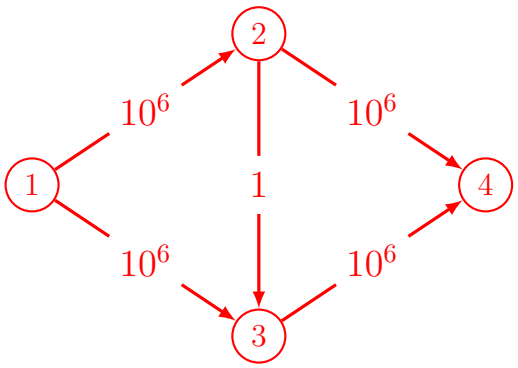
In the residual graph, C is the only node reachable from S. Thus the dotted line below is the min-cut. The forward edges traversing the cut are (S, A) and (C, D). Thus thus min-cut is 19, which is the same as the max flow.



2. Recall that the Ford-Fulkerson algorithm runs in $O(C|E|)$. Construct a flow network with $|E| \leq 10$ that requires at least 10^6 operations to complete.

Consider the flow network where $V = \{1, 2, 3, 4\}, E = \{(1, 2), (2, 4), (1, 3), (3, 4), (2, 3)\}$, $s = 1$, and $t = 4$. The capacity function has the following values: $c(1, 2) =$

$10^6, c(2,4) = 10^6$, $c(1,3) = 10^6, c(3,4) = 10^6$, and $c(2,3) = 1$. If we are unlucky with our choice of augmenting path, we can send only a unit of flow each path. This augmenting path will pass through the edge $(2,3)$ or the residual edge $(3,2)$. The maximum flow on this network is $2 \cdot 10^6$, and this means the we run $2 \cdot 10^6$ augmenting paths that take $O(m)$ operations to compute.



3. There are n students in a class. We want to choose a subset of k students as a committee. There has to be m_1 number of freshmen, m_2 number of sophomores, m_3 number of juniors, and m_4 number of seniors in the committee. Each student is from one of k departments, where $k = m_1 + m_2 + m_3 + m_4$. Exactly one student from each department has to be chosen for the committee. We are given a list of students, their home departments, and their class (freshman, sophomore, junior, senior). Describe an efficient algorithm based on network flow techniques to select who should be on the committee such that the above constraints are all satisfied.

   Algorithm: Reduce the committee assignment problem to max-flow as follows: We will construct a flow network G such that G can have a max flow of value k iff there is a feasible assignment of students to the committee. The construction will involve the following sets of nodes and edges:

   - 4 nodes representing freshman, sophomore, junior, and senior classes. n nodes representing students, k nodes representing departments, a node called s and a node called t
   - We will add directed edges from s to the four nodes representing freshman, sophomore, junior, and senior classes with capacities of m_1, m_2, m_3, and m_4 respectively. We will connect edges from each node representing a class to all students belonging to that class with capacity of 1. We will add edges from each student to the department they belong to with a capacity of 1. We will add edges from each department to t with a capacity of 1 .

   We will find max flow f in G and if $v(f) = k$, then we assert there is a valid assignment

of students to form the committee.

Explanation: If $v(f) = k$, it means that each edge from a department to t is saturated and also each edge from s to the four classes is saturated (since $k = m_1 + m_2 + m_3 + m_4$). Therefore, the constraints of (i) taking one students from each department and (ii) selecting an exact number of freshmen, sophomores, etc. are all met.

4. You want to assign n students to m project teams as evenly as possible. Each student has provided a list of projects he or she is willing to work on, and each project has at least one interested student. Ideally, the students would be evenly spread between the projects, so that no project takes more than $\lceil \frac{n}{m} \rceil$ students. However, this will likely be impossible. Design a polynomial-time algorithm to find an assignment that minimizes the number of students assigned to each project team.

   Hints

   - Construct a graph in $\Theta(mn)$ time. You can leave some of the edge capacities undetermined.
   - Assume that each project will take no more than n students. Is it possible that the value of the max-flow is lower than n?
   - Assume that each project will take no more than $\lceil \frac{n}{m} \rceil$ students. Is it possible that the value of the max-flow is lower than n?
   - Run Ford-Fulkerson algorithm at most $\Theta(\log n)$ times.

   (a) Construct a graph in the following way

   - Add a source node s and a sink node t
   - Add a node s_i for each student i
   - Add edges of capacity 1 from s to each student node s_i
   - Add a node p_j for each project j
   - Add edges of capacity x from each project node p_j to t
   - For each student node s_i, add edges of capacity 1 from s_i to each project p_j that the student is interested in

   (b) Initially, set $x = \lceil \frac{n}{m} \rceil$. Run Ford-Fulkerson algorithm to find a max-flow on G. If a flow of n is found, then we are done. Otherwise, we should increase x and run Ford-Fulkerson algorithm again. Perform a binary search on x between its minimum value of $\lceil \frac{n}{m} \rceil$ and its maximum value of n until the smallest value of

x for which a flow of n is possible has been found. Construct the assignment according to the flow.

(c) Time complexity: $\Theta(n^2 m \log n)$

5. We define a most vital edge of a network as an edge whose deletion causes the largest decrease in the maximum s-t-flow value. Let f be an arbitrary maximum s-t-flow. Either prove the following claims or show through counterexamples that they are false:

   (a) A most vital edge is an edge e with the maximum value of c_e.

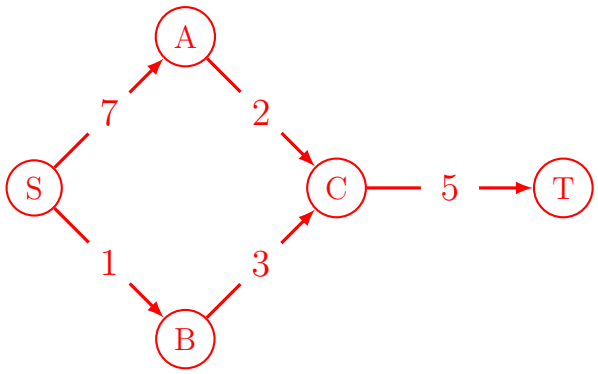
   (b) A most vital edge is an edge e with the maximum value of f_e.

   (c) A most vital edge is an edge e with the maximum value of f_e among edges belonging to some minimum cut.

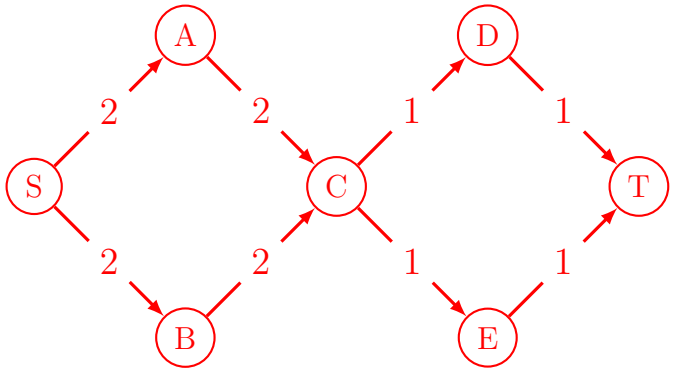
   (d) An edge that does not belong to any minimum cut cannot be a most vital edge.

   (e) A network can contain only one most vital edge.

(a) False



(b) False



(c) False: Same as (a)

(d) False: Same as (a)

(e) False: