COMPSCI 311: Introduction to Algorithms **Fall 2022**

Discussion 9

Your Name: ____________________ Collaborators: ______________________________________

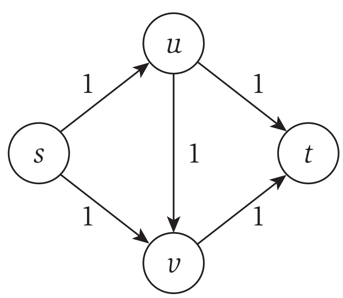
Definitions.

1. An s-t cut (A, B) is a partition of the nodes into sets A and B where $s \in A$, $t \in B$
2. The *capacity* of cut (A, B) equals

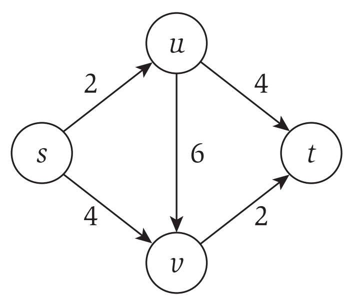
$$c(A, B) = \sum_{e \text{ from } A \text{ to } B} c(e)$$

3. A *minimum* cut is an s-t cut with minimum capacity.

Problem 1. Flow Network.



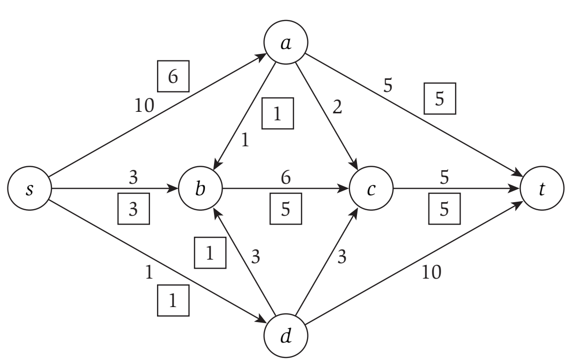
(a)



(b)

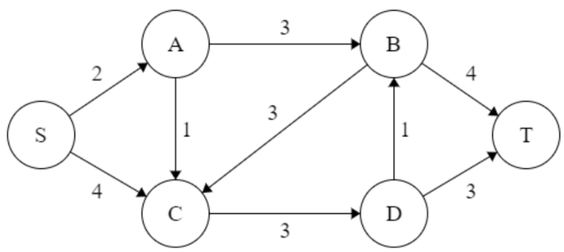
1. List all the minimum s-t cuts in flow network (a) above. The capacity of each edge appears as a label next to the edge.
2. What is the minimum capacity of an s-t cut in flow network (b) above? Again, the capacity of each edge appears as a label next to the edge.

Problem 2. Flow Network. The following figure shows a flow network on which an s-t flow has been computed. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge. (Edges without boxed numbers have no flow being sent on them.)



1. What is the value of the flow? Is this a maximum (s-t) flow in this graph?
2. Find a minimum s-t cut and also say what its capacity is.

Problem 3. Flow Network. In the flow network illustrated below, each directed edge is labeled with its capacity. We are using the Ford-Fulkerson algorithm to find the maximum flow. The first augmenting path is $S - A - C - D - T$, and the second augmenting path is $S - A - B - C - D - T$.



1. Draw the residual network after we have updated the flow using these two augmenting paths (in the order given).
2. List all of the augmenting paths that could be chosen for the third augmentation step.
3. Draw a line through the original graph to represent the minimum s-t cut and also say what its capacity is.

Problem 4. Train Tickets (more DP practice, if time). You want to travel from stop 1 to stop n on a railway line with n stops. You can buy a ticket from stop j to stop k for c_{jk} dollars for any $j < k$. What is the cheapest way to get from stop 1 to stop n?

Define OPT(j) to be the cheapest way to get from stop j to stop n. Write a recurrence for OPT(j) and include a base case. (**Hint**: what is the first decision to make when deciding how to get from stop j to stop n? What are the possible choices for the first decision? Write a recurrence that has one case for each choice.)

Write an iterative algorithm to compute OPT(1), the cost to go from stop 1 to n. What size is the memoization array? In what order should array entries be filled? What is the running time?