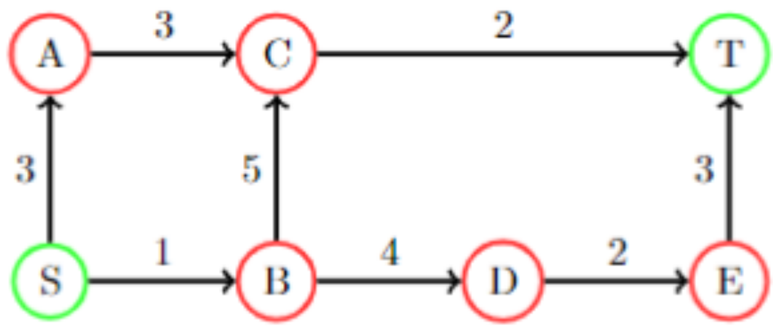
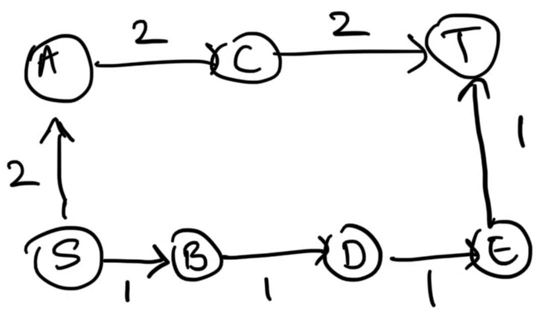
HW6

Problem 1
The following graph G has labeled nodes and edges between them. Each edge is labeled with its capacity.

(a) Draw the final residual graph G_f using the Ford-Fulkerson algorithm corresponding to the max flow. Please do NOT show all intermediate steps.
(b) What is the max-flow value?
(c) What is the min-cut?



Solution: Final Residual Graph :



Max Flow = 3
Min Cut = {(C, T), (B, C), (S, B)}

Problem 2
Determine if the following statements are true or false. For each statement, briefly explain your reasoning.

(a) In a flow network, the value of flow from S to T can be higher than the maximum number of edge-disjoint paths from S to T. (Recall that edge-disjoint paths are paths that do not share any edge)

(b) For a flow network, there always exists a maximum flow that doesn't include a cycle containing positive flow.

(c) If you have non-integer edge capacities, then you cannot have an integer max-flow value.

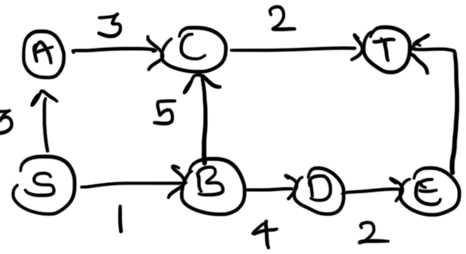
(d) Suppose the maximum s-t flow of a graph has value f. Now we increase the capacity of every edge by 1. Then the maximum s-t flow in this modified graph will have a value of at most f +1.

(e) If all edge capacities are multiplied by a positive number k, then the min-cut remains unchanged.

Solution:

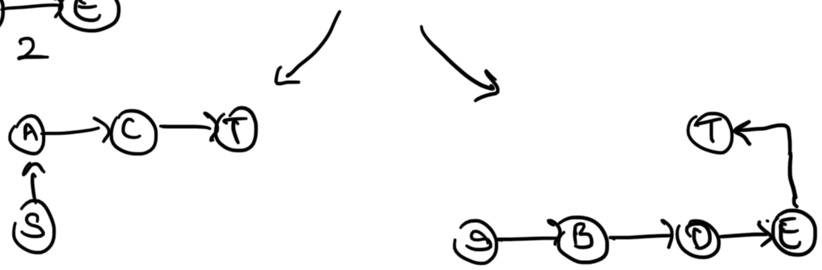
(a)

True.

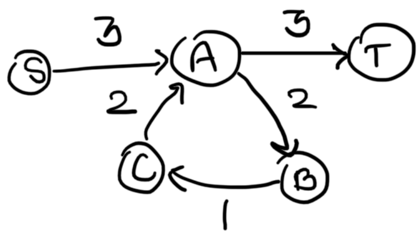


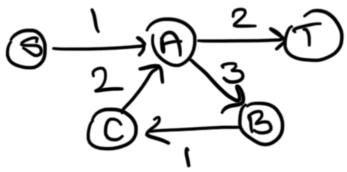
max flow = 3

max no. of edge disjoint paths = 2



ⓑ True.





The flow in the first network is 1.

If we decide to remove the cycle

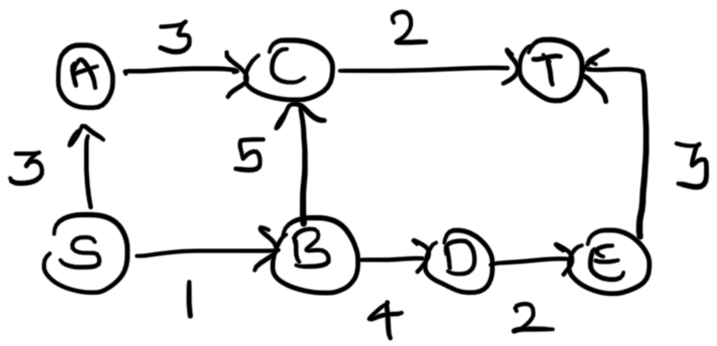
{(A,B),(B,C),(C,A)}, our max flow increases.

In the second diagram however the flow remains the same. Therefore the existence of a cycle leads to no increase in flow and thus a max flow can always exist without a cycle.

(c) False.

Edge capacities in a flow network don't need to be integers for the maximum flow to be an integer. According to the Max-Flow Min-Cut theorem, the maximum flow through a network is equivalent to the network's minimum cut. Consequently, if the minimum cut in the network is an integer, the maximum flow will also be an integer, irrespective of whether the capacities are integers. Therefore, having non-integer edge capacities does not inherently mean a non-integer maximum flow.

(d) False.



In the network above, if all the edge values were increased by 1, then the maximum flow (f) would go from 3 to 5 = f + 2, thus contradicting the statement.

(e) False.

The minimum cut (min-cut) in a network refers to the lowest total capacity of edges that, when removed, disconnects the source and the sink. If every edge capacity in the network is scaled by a positive constant k, then the min-cut value will also be scaled by this same factor. This is a result of the min-cut's direct correlation with the capacity values of the network's edges. So, the min-cut doesn't stay constant, instead, it increases or decreases in proportion to the scaling factor applied to the capacities of the edges.

Problem 3

You are given a flow network with unit-capacity edges. It consists of a directed graph G=(V,E) with source s and sink t, and u_e = 1 for every edge e. You are also given a positive integer parameter k. The goal is delete k edges so as to reduce the maximum s-t flow in G by as much as possible. In other words, you should find a subset of edges F $\subseteq$ E such that |F| = k and the maximum s-t flow in the graph G' = (V,E \F) is as small as possible. Give a polynomial-time algorithm to solve this problem.

Follow up: If the edges have more than unit capacity, will your algorithm produce the smallest possible max-flow value?

Solution: Consider that when we remove k edges from a graph, the capacity of any cut can decrease by at most k, which means the minimum cut value will also decrease by at most k. Consequently, the maximum flow will decrease by at most k. We can go further to prove that it is indeed possible to reduce the max-flow by exactly k. To achieve this, we select a minimum cut X and remove k edges that emanate from this cut. As a result, the capacity of the cut will be reduced to f - k, where f represents the value of the max-flow. Thus, the min-cut will be f - k, leading to the max-flow becoming f - k.

No, the algorithm described will not necessarily produce the smallest possible max-flow value when the edges have capacities greater than one. When the edges have capacities greater than one, removing k edges may not always lead to the smallest possible max-flow value.

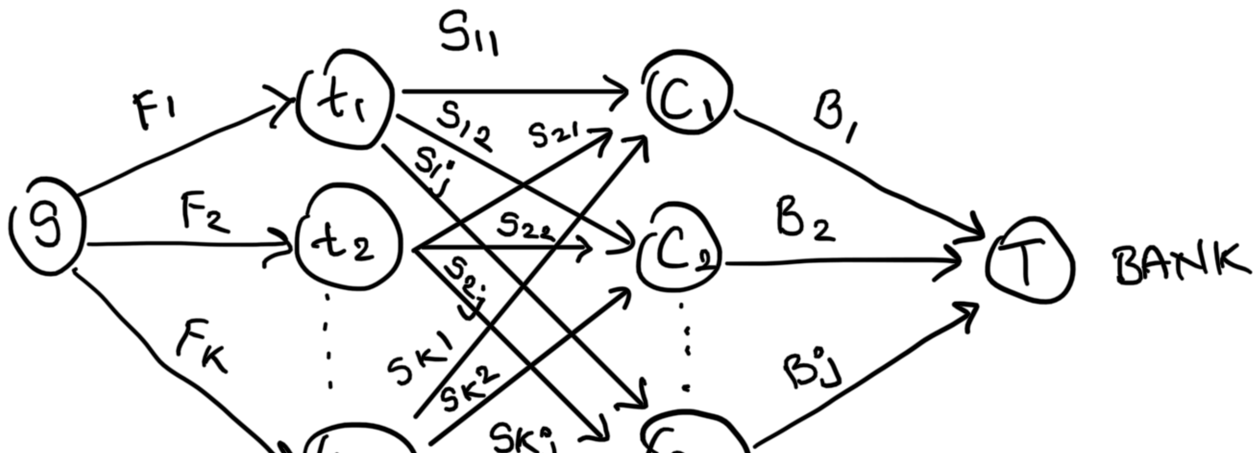
<u>Problem 4</u>

A tourist group needs to convert all of their USD into various international currencies. There are n tourists $t_1, t_2, ..., t_n$ and m currencies $c_1, c_2, ..., c_m$. Each tourist t_k has F_k Dollars to convert. For each currency c_j, the bank can convert at most B_j Dollars to c_j. Tourist t_k is willing to trade at most S_{kj} of their Dollars for currency c_j. (For example, a tourist with 1000 dollars might be willing to convert up to 300 of their USD for Rupees, up to 500 of their USD for Japanese Yen, and up to 400 of their USD for Euros). Assume that all tourists give their requests to the bank at the same time.

(a) Design an algorithm that the bank can use to determine whether all requests can be satisfied. To do this, construct and draw a network flow graph, with appropriate source and sink nodes, and edge capacities.

(b) Prove your algorithm is correct by making an if-and-only-if claim and proving it in both directions.

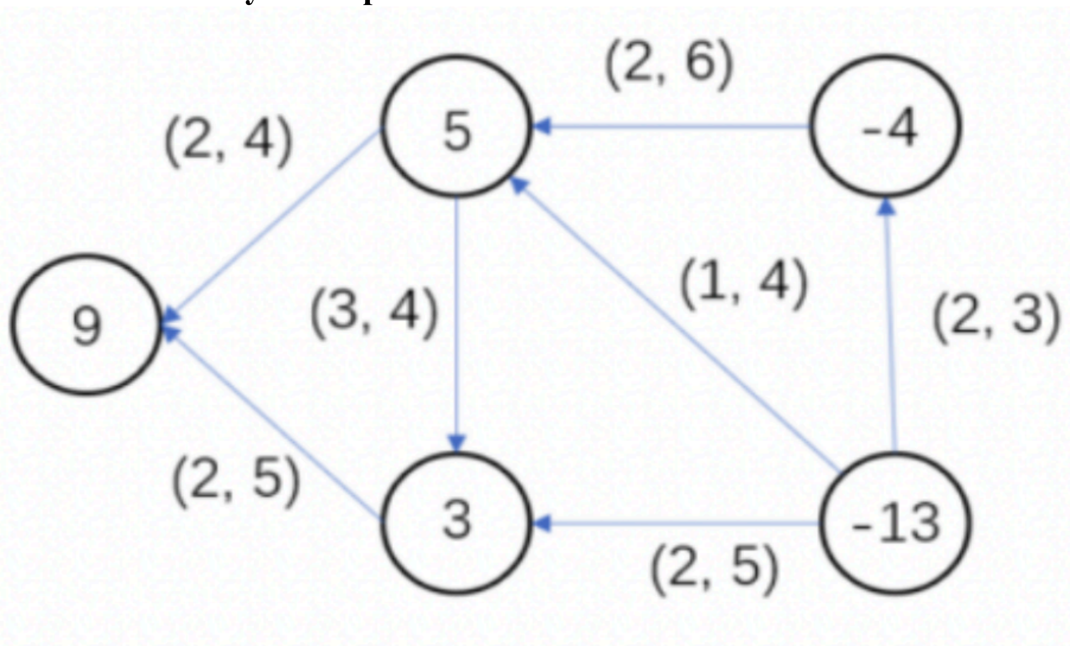
Solution: (a)



(b) In order to validate the accuracy of this algorithm, we have to demonstrate that the conversion requests can be fulfilled if and only if the maximum flow aligns with the total amount of dollars owned by the tourists. We can assert that all the conversion requests are met if the maximum flow equates to the total sum of dollars possessed by the tourists, as this would signify that all conversions have successfully occurred, allowing the full flow to accumulate at the sink. On the other hand, if the maximum flow is indeed equivalent to the total sum of dollars held by all the tourists, we can infer that all the requests have been satisfied. This is because a similar reasoning can be applied: all the dollars have been successfully converted and have congregated at the sink.

Problem 5

In the network G below, the demand values are shown on vertices (supply value if negative). Lower bounds on flow and edge capacities are shown as (lower bound, capacity) for each edge. Determine if there is a feasible circulation in this graph. You need to show all your steps.



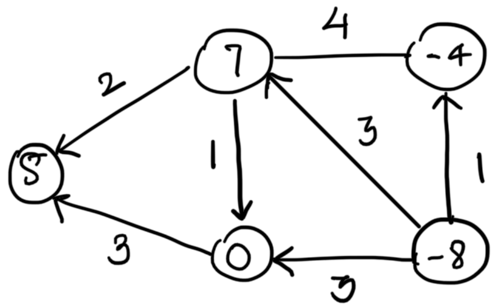
(a) Reduce the Feasible Circulation with Lower Bounds problem to a Feasible Circulation problem without lower bounds.

(b) Reduce the Feasible Circulation problem obtained in part (a) to a Max Flow problem in a Flow Network.

(c) Solve the resulting Max Flow problem and explain whether there is a Feasible Circulation in the original G.

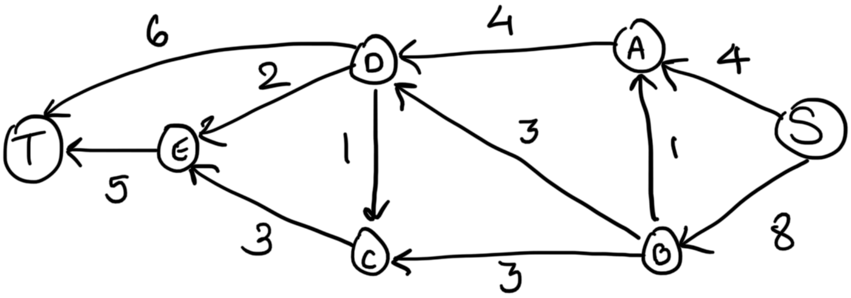
Solution:

(a)



sum of all demands = 12

(b)



(c) Max flow = 10

There is no feasible circulation since max flow ≠ sum of all demands.