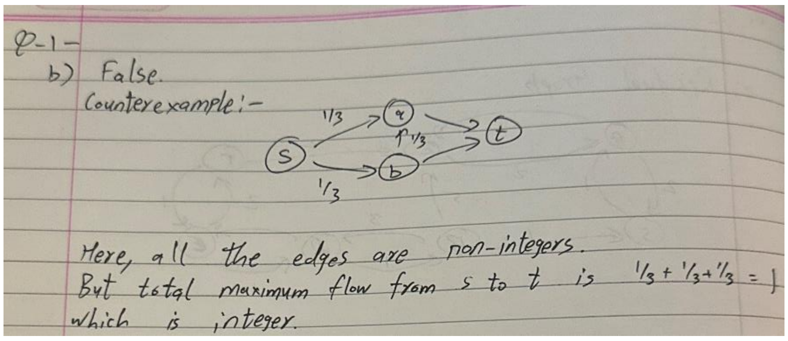
Homework 8

1. **Determine if the following statements are true or false. If true, give a brief explanation. If false give a counterexample. (10 points)**

a) **For a flow network, there always exists a maximum flow that doesn't include a cycle containing positive flow.**
Ans:
True.
There always exist a maximum flow that doesn't including a cycle containing positive values. This is because if such cycle exists, we can augment the flow by sending more flow in that cycle, which increases the total flow in the network. But since its already a maximum flow, there is no way this happens.

b) **If you have non-integer edge capacities, then you cannot have an integer max-flow value**.
Ans:
False.

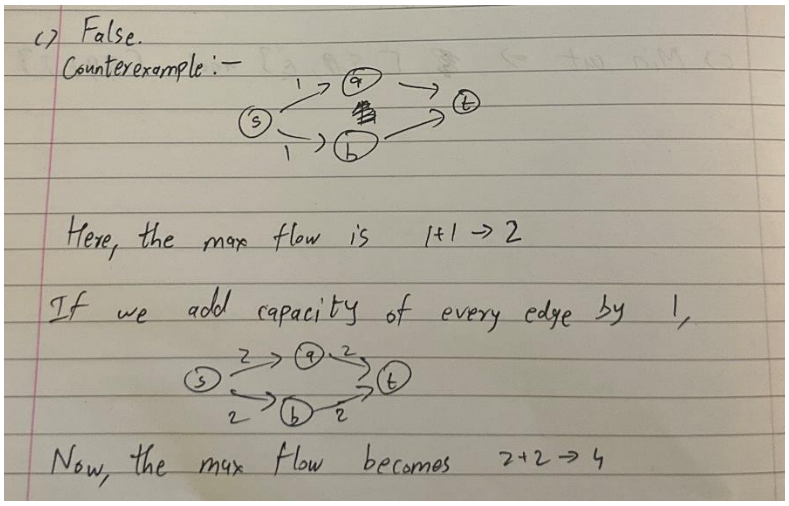


Q-1-
b) False.
Counterexample:-

$$\begin{tikzcd}
 & a \arrow[dr] & \\
s \arrow[ur, "1/3"] \arrow[dr, "1/3"'] & & t \\
 & b \arrow[uu, "1/3"] \arrow[ur] &
\end{tikzcd}$$

Here, all the edges are non-integers.
But total maximum flow from s to t is $1/3 + 1/3 + 1/3 = 1$
which is integer.

c) **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.**
Ans:
False.



c) False.
Counterexample:-

$$\begin{tikzcd}
 & a \arrow[dr] & \\
s \arrow[ur, "1"] \arrow[dr, "1"'] & & t \\
 & b \arrow[ur] &
\end{tikzcd}$$

Here, the max flow is $1+1 \rightarrow 2$

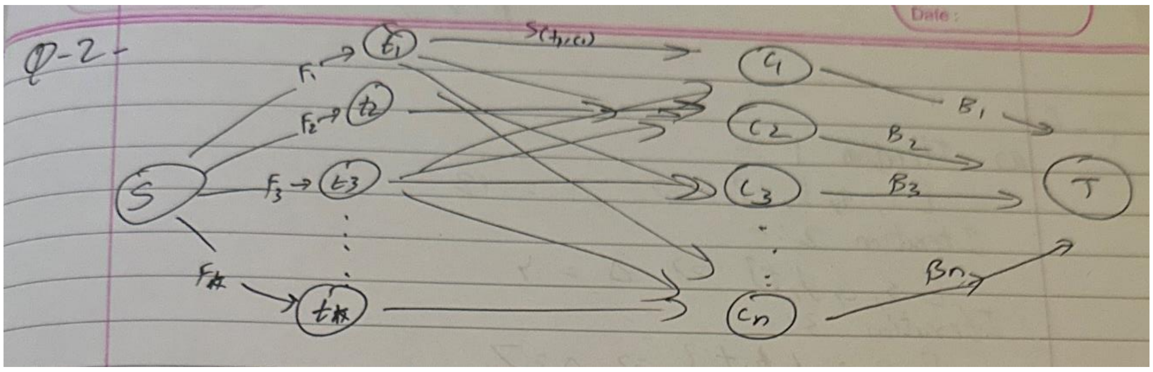
If we add capacity of every edge by 1,

$$\begin{tikzcd}
 & a \arrow[dr, "2"] & \\
s \arrow[ur, "2"] \arrow[dr, "2"'] & & t \\
 & b \arrow[ur, "2"'] &
\end{tikzcd}$$

Now, the max flow becomes $2+2 \rightarrow 4$

d) **If all edge capacities are multiplied by a positive number k , then the min-cut remains unchanged.**
Ans:
True.
The min-cut depends on the capacities of the edges, and not their absolute value. When we multiply all edges by k, it scales the capacity of each edge uniformly without changing the relative relationships between them. Hence, the relative capacities between edges are unchanged.

2. **A tourist group needs to convert all of their USD into various international currencies. There are n tourists t1,t2, ...,tn and m currencies c1,c2, ...,cm. Each tourist tk has Fk Dollars to convert. For each currency cj , the bank can convert at most Bj Dollars to cj. Tourist tk is willing to trade at most Skj of their Dollars for currency cj. (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. 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. Prove your algorithm is correct by making an if-and-only-if claim (10 points)**
Ans:



Here, S is the source node connected to each tourist node with an edge of capacity Fk. This tourist node is connected to each currency node cj with an edge of capacity Skj (tk and cj). Each currency node with then connected to sink node T with an edge capacity of Bj.

Algorithm:
Construct the network flow.
Apply Ford-Fulkerson Algorithm to find the max flow from S to T.
IF max flow equals to the total amount of USD the tourist have, then all requests can be satisfied. Else, requests cannot be fulfilled.
End IF

Proof (If and only if claim):
Claim:
All requests can be satisfied if and only if the maximum flow in the constructed network flow graph equals the total amount of USD the tourists have.

Proof:
Forward Claim: If all the requests are satisfied, then the max flow in the graph equals the total amount of USD the tourists have. In the constructed network flow graph, there should be a flow from S to T that saturates all edges leaving S and entering T. Therefore, the maximum flow in the graph equals the total amount of USD the tourists have.

Backward Claim: If the maximum flow in the graph equals the total amount of USD the tourists have, this means that there exists a flow from S to T that saturates all edges leaving S. This means that all tourist requests can be satisfied without exceeding the bank's capacity to convert USD to different currencies.
Thus, all requests can be satisfied.

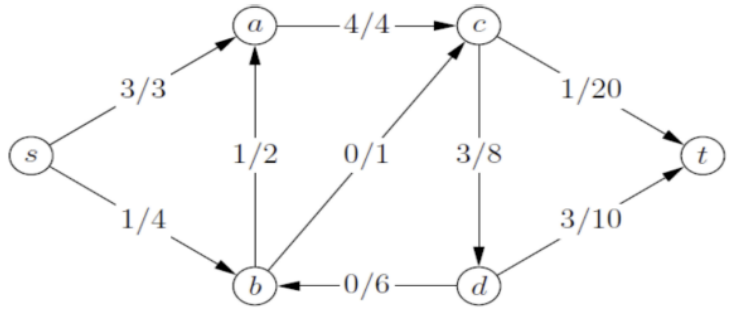
3. **You are given a collection of n points U={u1,...,un} in the plane, each of which is the location of a cell-phone user. You are also given the locations of m cell-phone towers,C={c1,...,cm}. A cell-phone user can connect to a tower if it is within distance Δ of the tower. For the sake of fault-tolerance each cell-phone user must be connected to at least three different towers. For each tower ci you are given the maximum number of users, mi that can connect to this tower. Give a polynomial time algorithm, which determines whether it is possible to assign all the cell-phone users to towers, subject to these constraints. Prove your algorithm is correct by making an if-and-only-if claim. (You may assume you have a function that returns the distance between any two points in O(1) time.) (10 points)**
Ans:
Algorithm:
Construct a bipartite graph in which one set of vertices represents the cell-phone users, and other set represents the cell-phone towers.
Connect every user to the towers within Δ distance.
Use Edmonds-Karp algorithm to find the maximum matching in the graph.
If the maximum matching covers all users, return “Possible”, Else “Impossible”.
End IF.

Proof of Correctness (if-and-only-if claim):
Claim: It is possible to assign all cell-phone users to towers subject to the given constraints if and only if the maximum matching in the constructed bipartite graph covers all users.

Proof:
Forward Claim: If it is possible to assign all cell-phone users to towers subject to the given constraints, this means that there exists a way to connect each user to at least three different towers within distance Δ. In the bipartite graph, this translates to a maximum matching that covers all users.

Backward Claim: If the maximum matching in the graph covers all users, this means that each user is connected to at least one tower within distance Δ. Since each user must be connected to at least three different towers, and the maximum matching covers all users, it means that each user can indeed be connected to at least three different towers.
Therefore, it is possible to assign all cell-phone users to towers subject to the given constraints.

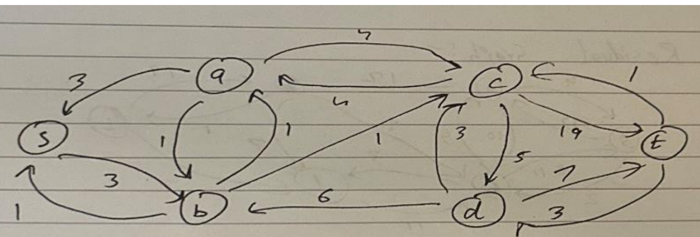
4. **You are given a directed graph which after a few iterations of Ford-Fulkerson has the following flow. The labeling of edges indicate flow/capacity: (15 points)**



a) **Draw the corresponding residual graph.**
b) **Is this a max flow? If yes, indicate why. If no, find max flow.**
c) **What is the min-cut?**

Ans:

Q-4- a)



b) No, this is not the max flow.

Max flow :-

$s \rightarrow b \rightarrow c \rightarrow t$

3 1 1 $\longrightarrow$ 3+1+1=5

c) Min cut :-

{s, a} and {b, c, d, t}

5. **USC has resumed in-person education after a one-year break, with k on-site courses available this term, labeled c1 through ck . Additionally, there are n students, labeled s1 to sn, attending these k courses. It's possible for a student to attend multiple on-site courses, and each course will have a variety of students enrolled. (20 points).**
   a) **Each student sj wishes to enroll in a specific group pj of the k available courses, with the condition that each must enroll in at least m courses to qualify as a full-time student (where pj is greater than or equal to m). Furthermore, every course ci can only accommodate a maximum of qi students. The task for the school's administration is to verify whether every student can register as a full-time student under these conditions. Propose an algorithm to assess this scenario. Prove your algorithm is correct by making an if-and-only-if claim.**

      Ans:

      Algorithm:

      Construct a bipartite graph in which one set of vertices represent students (labelled s1 to sn) and other edge represents courses (c1 to ck).

      Connect each student to all the courses in the group pj.

      Assign weights to the edges based on the accommodation of each course qi.

      If a course already has qi students, then weight becomes 0.

      Use Ford-Fulkerson algorithm to find the maximum matching in the graph.

      If every student is not matched to at least m courses, the algorithm terminates with the result that not every student can register as a full-time student. Else, check if the maximum matching satisfies the capacity constraint for each course. If any course has more than qi students assigned to it, the algorithm terminates with the result that not every student can register as a full-time student.

      If all conditions are satisfied, the algorithm terminates with the result that every student can register as a full-time student.

      Proof of Correctness (if-and-only-if claim):

      Claim: Every student can register as a full-time student if and only if the algorithm terminates with the result that every student can register as a full-time student.

      Proof:

      Forward claim: If every student can register as a full-time student, it means that each student is assigned to at least m courses. Additionally, each course has a maximum capacity of qi students. By constructing the bipartite graph and assigning weights based on the available capacity of each course, the maximum matching algorithm will ensure that no course is overbooked, satisfying both conditions. Therefore, the algorithm will terminate with the result that every student can register as a full-time student.

      Backward Claim: If the algorithm terminates with the result that every student can register as a full-time student, it means that every student is matched to at least m courses, and no course is overbooked. This implies that every student can indeed register as a full-time student while satisfying both conditions.

      Hence, the claim is proved, and the algorithm is correct.

b) **Assuming a viable solution is found for part (a) where each student is enrolled in exactly m courses, there arises a need for a student representative for each course from among the enrolled students. However, any single student can represent at most r (where r is less than m) courses in which they are enrolled. Develop an algorithm to check whether it is possible to appoint such representatives, building on the solution from part (a) as a foundation. Prove your algorithm is correct by making an if-and-only-if claim.**

Ans:

Algorithm:

Continuing the solution from (a), use the same nodes with exactly m courses that have been enrolled by each student.

Assign 1 as the capacity from classes to sink

Assign r for edges between source and student vertices.

Use Ford-Fulkerson algorithm to find the maximum matching in the graph.

If every student is not matched to at least m courses, the algorithm terminates with the result that not every student can register as a full-time student. Else, check if the maximum matching satisfies the capacity constraint for each course. If any course has more than qi students assigned to it, the algorithm terminates with the result that not every student can register as a full-time student.

If all conditions are satisfied, the algorithm terminates with the result that every student can register as a full-time student.

Proof of correctness(If and only if claim):

Claim: It is possible to appoint student representatives for each course, ensuring that each student represents at most r courses, if and only if the maximum matching in the constructed bipartite graph satisfies the additional constraint.

Proof:

Froward Claim:

If it is possible to appoint student representatives for each course, ensuring that each student represents at most r courses, then this implies that each course is represented by at most r students. In the constructed bipartite graph, this translates to a maximum matching that satisfies the additional constraint.

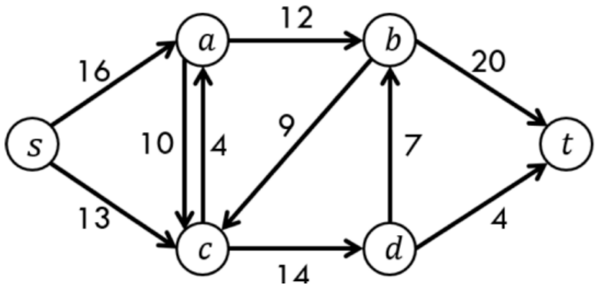
Backward claim:

If the maximum matching in the graph satisfies the additional constraint, it implies that each course is represented by at most r students.

Since each student can represent at most r courses, and the maximum matching satisfies the additional constraint, it means that it is possible to appoint student representatives for each course, ensuring that each student represents at most r courses.

Therefore, the algorithm correctly checks whether it is possible to appoint student representatives for each course, ensuring that each student represents at most r courses, by examining if the maximum matching in the constructed bipartite graph satisfies the additional constraint.

6. **Given the below graph solve the below questions using scaled version of Ford-Fulkerson. (15 points)**



a) **Give the Δ and path selected at each iteration.**
b) **Draw the final network graph and the residual graph**
c) **Find the maxflow and mincut**

Ans:

Q-6-

a) Iteration 1 :-

[s, a, b, t] => Δ = 12

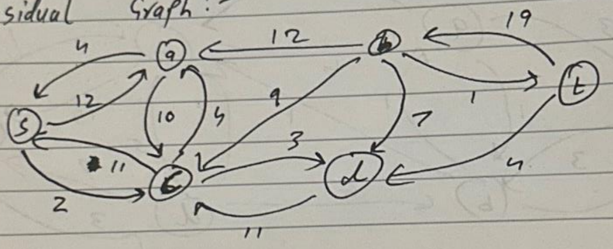
Iteration 2 :-

[s, c, d, t] => Δ = 4

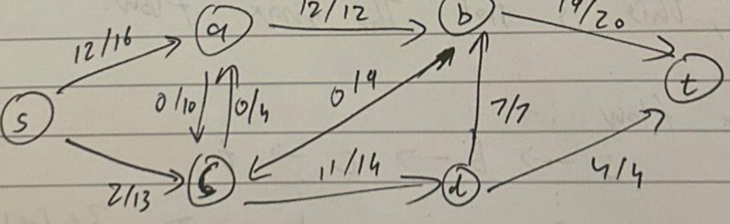
Iteration 3 :-

[s, c, d, b, t] => Δ = 7

b) Residual Graph :-



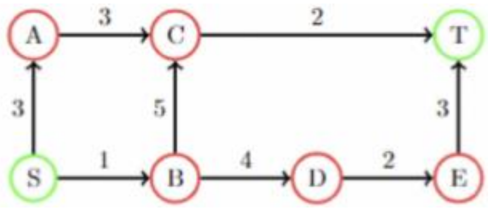
Network Graph :-



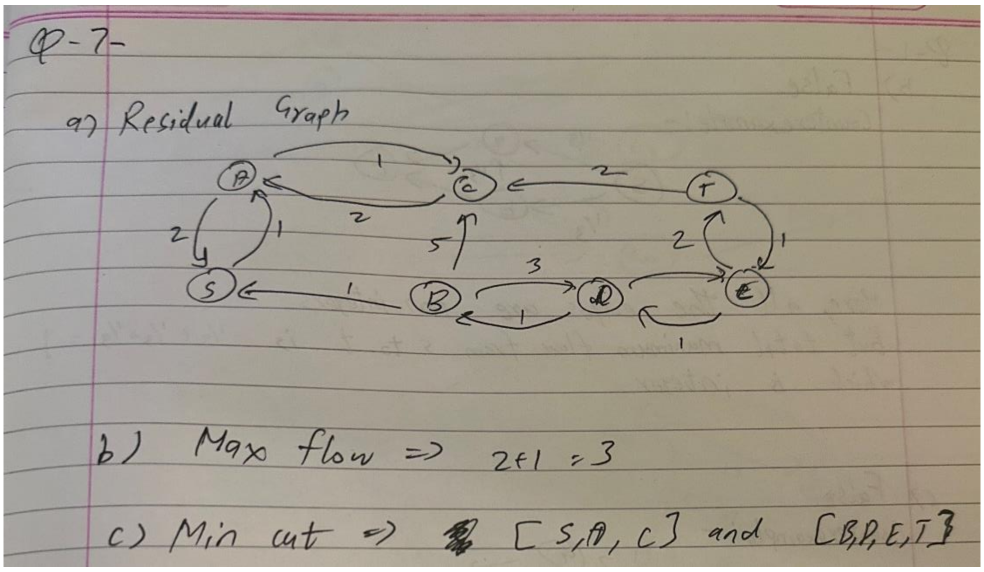
c) Max Flow => 12+ 4+7 = 23

Min cut => [s, a, c, d] and [b, t]

7. **The following graph G has labeled nodes and edges between them. Each edge is labeled with its capacity. (10 points)**



(a) Draw the final residual graph Gf 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?
Ans:



Q-7-

a) Residual Graph

b) Max flow => 2+1 = 3

c) Min cut => [S, A, C] and [B, D, E, T]

8. **You are provided with a flow network where each edge has a capacity of one. This network is represented by a directed graph G = (V, E), including a source node s and a target node t. Additionally, you are given a positive integer k. The objective is to remove k edges to achieve the greatest possible reduction in the maximum flow from s to t in G. Your task is to identify a subset of edges F within E, where the size of F equals k, and removing these edges from G results in a new graph G' = (V, E-F) where the maximum flow from s to t is minimized. Propose a polynomial-time strategy to address this issue.**
Furthermore, consider if the capacities of the edges are greater than one, and discuss whether your strategy still ensures the lowest possible maximum flow. (20 points)
Ans:
Here, we use min cut.
Polynomial-time strategy:
Use Edmond-Karps algorithm to find the maximum flow in the flow network.
Use the residual graph obtained from the maximum flow algorithm to find the minimum cut in the network.

Identify the k edges with the smallest capacities in the minimum cut.
Remove these k edges from the original graph G to obtain the new graph G' = (V, E-F).
Return G' as the result.

Regarding the capacities of the edges being greater than one, the strategy still ensures the lowest possible maximum flow. This is because the essence of the strategy lies in identifying and removing the edges with the smallest capacities in the minimum cut. Regardless of the original capacities of these edges, their removal diminishes the overall capacity of the minimum cut, leading to a reduction in the maximum flow from s to t. Thus, the strategy effectively achieves its objective even when dealing with edges of varying capacities.