ADELINE MAKOKHA 25/01/2025

ADM NO 191199 CAT 2

DSA 8205 : OPTIMIZATION FOR DATA SCIENCE

Question One (10mks)

i) State the Ford - Fulkerson theorem for network flows.

It states that in a flow network, the maximum flow from the source s to the sink t is equal to the capacity of the minimum cut separating s and t that is

Max flow = Min cut.

Explain how the network flow algorithm may be used to prove it for integer capacities.

- The Ford - fulkerson algorithm iteratively augments the flow along augmenting paths in the residual graph until no such paths exist.
- At each step, the algorithm finds a path from s to t with positive residual capacity and increases the flow along this path by the minimum residual capacity on the path.
- Since capacities are integers, each augmentation increases the flow by at least 1. This ensures the algorithm terminates in a finite number of steps.
- When no augmenting paths remain, the algorithm terminates, and the flow is maximal. The set of nodes reachable from s in the residual graph defines the minimum cut, whose capacity equals the maximum flow.

Indicate what problem you may encounter if you allowed non - integer flows and capacities.

For non - integer capacities are being allowed, the Ford - Fulkerson algorithm may fail to terminate or converge to the correct maximum flow.

This is because the algorithm might make small augmentations, leading to an infinite loop.

What theorem in analysis makes the Ford - Fulkerson theorem still valid.
The maximum - flow Min-cut theorem still holds for non-integer capacities. This is guaranteed by the completeness Axiom of real numbers in real analysis, which ensures that the algorithm converges to the maximum flow even in the presence of non-integer capacities.
For integer capacities, the Ford - Fulkerson algorithm terminates and proves the theorem directly. For non-integer capacities, the theorem is still valid due to the properties of real numbers and the Edmonds - karp algorithm.

iii) Constructing the Flow Network.

A bipartite $G = (L \cup R, E)$ be a bipartite graph, where L and R are the two disjoint sets of vertices and E is the set of edges.

2. Flow Network Construction

Add a source node s and connect it to all vertices in L with edges of capacity 1.

Add a sink node t and connect all vertices in R to t with edges of capacity 1.

Direct all original edges E' from L to R and assign them a capacity of 1.

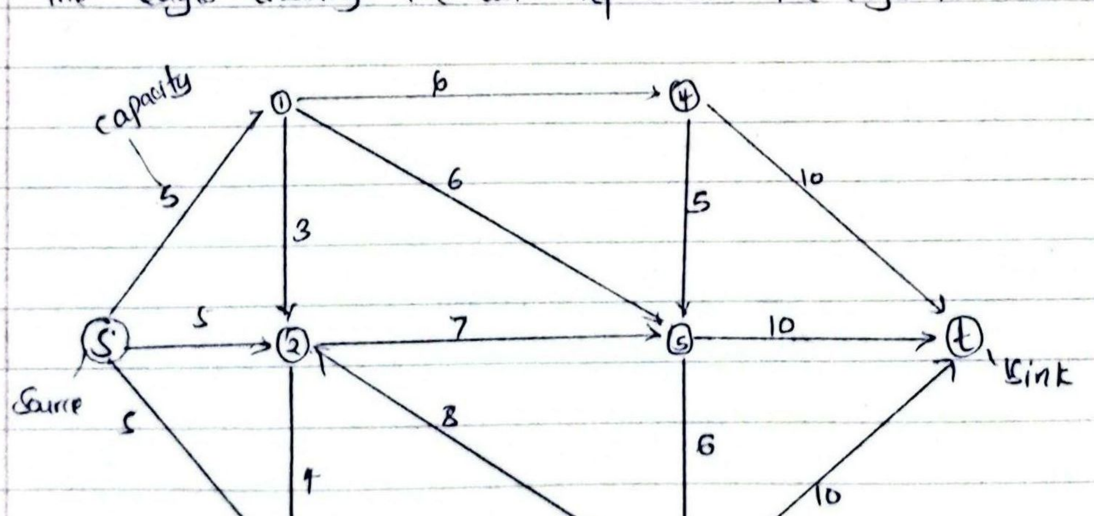
The resulting flow network is $G' = (V', E')$, where $V' = L \cup R \cup \{s, t\}$ and E' includes the new edges from s to L, from R to t, and the original edges from L to R

3. Maximum Flow and Maximum Matching:

A maximum flow in G' corresponds to a maximum matching in G. Each unit of flow from s to t represents an edge in the matching, and the flow conservation ensures that no two edges in the matching share a vertex.

4. Minimum Cut and Minimum Edge Cover.

A minimum cut in G' corresponds to a minimum edge cover in G. The cut partitions the graph into two sets S and T, where $s \in S$ and $t \in T$. The edges crossing the cut represent the edges in the edge cover.



$G(L, R, E)$

(L, R) - Vertices $\quad$ E - Edges.

Prove that the maximum size of a matching is the same as the minimize size of a cover in a bipartite graph.

A matching is a set of edges with no shared vertices

An edge cover is a set of edges such that every vertex is incident to at least one edge in the set.

For any matching M and edge cover C, then $|M| \leq |C|$. This is because each edge is M covers exactly two vertices, while each edge in C covers at least one vertex

Let M be a maximum matching. The vertices not covered by M must be covered by additional edges to form an edge cover. Add one edge for each uncovered vertex to M to form an edgecover C

Since M is maximal, the number of uncovered vertices is $|V| - 2|M|$, so

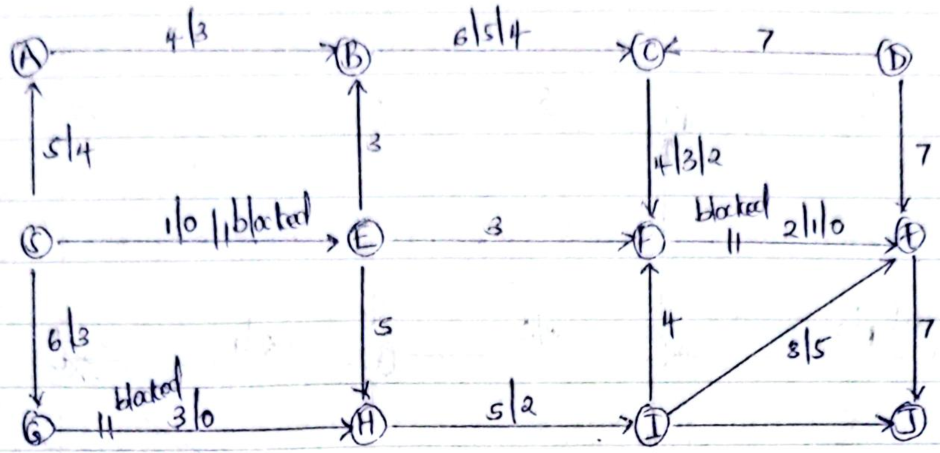
$|C| = |M| + (|V| - 2|M|) = |V| - |M|$

For a bipartite graph, the size of the minimum edge cover C is $|C| = |V| - |M|$, and the size of the maximum matching M is $|M| = |V| - |C|$. Thus, $|M| = |C|$

The maximum size of a match M equals the minimum size of an edge cover C in a bipartite graph.

Question Two (10 mks)

i) Consider the network N below with capacities on the edges.



Path	flow.
S→E→B→C→F→t	1
S→A→B→C→F→t	1
S→G→H→I→t	3
Total	5

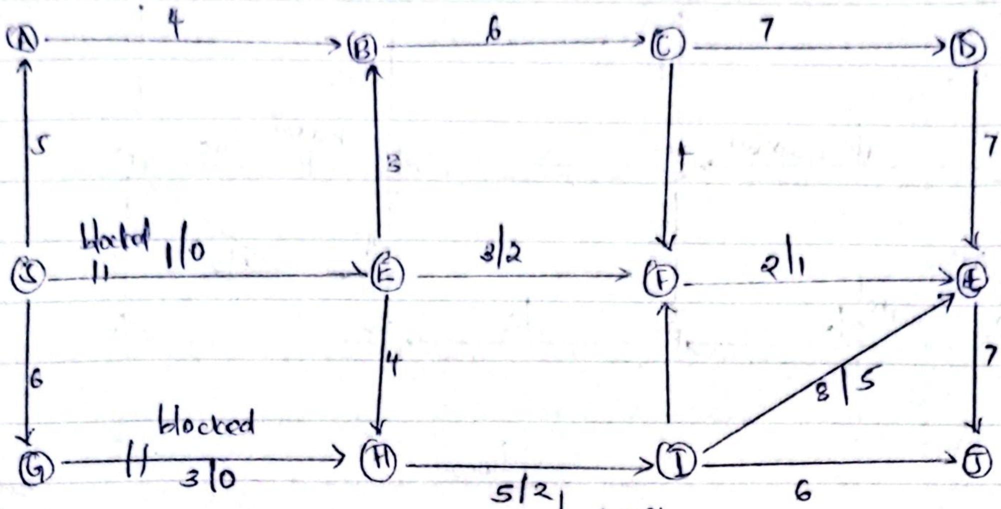
Max flow = 5

Min cut = Max flow

Removed edges	Flow
G→H	3
F→t	2
Total.	5

Min cut = 5

ii) Using Edmonds - karp's algorithm. to compute a maximum ow in the network N. For each augmenting path write the node on the path and the value you augment the path with in a table



Path	Mflow	
$S \to E \to F \to t$	min $\{1, 3, 2\}$	1
$S \to G \to H \to I \to t$	min $\{3, 5, 8\}$	3
$S \to A \to B \to C \to F \to t$	min $\{5, 4, 6, 4, 1\}$	1
Total		5

This is a specific implementation of the Ford- Fulkerson method for computing the maximum flow in a flow network.
Breadth - first Search (BFS) to systematically find augmenting paths and guarantee a polynomial time using the shortest augmenting path.