43159

ASSIGNMENT-6

TITLE : FORD FULKERSONS ALGORITHM

PROBLEM STATEMENT: Write a program for Ford Fulkerson's maximal flow algorithm

THEORY:

• FORD FULKERSON'S :

- Maximum flow problem involves finding a feasible solution flow through a single-source N/w, i.e., maximum.

- Each edge is labelled with capacity, the max stuff it can carry. We have to figure out the maximum stuff that can be pushed from vertex S (source) to vertex E (sink)

- This algorithm works by extending typical greedy algorithm by allowing undo operation.

• ALGORITHM:

i. Start with initial flow = 0

ii. While there is an augmenting path from source to sink, add this path flow to flow

iii. Return flow

- Time complexity = O(max flow * E)

• WORKING :

i. Initial residual capacity :
flow = 0 & graph = actual graph

43159

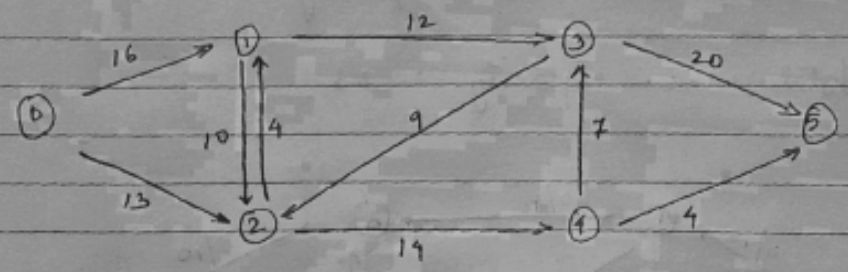
ii. BFS / DFS of residual graph to find a path that exists to sink

iii. Build a parent path array which we can extend & find min. residual cap.

iv. We add this path flow to overall flow

ex:

Q.1.

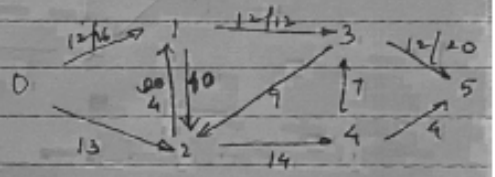


max. flow = 0

use BFS

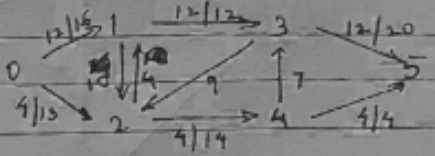
1. 0 → 1 → 3 → 5

min cap = 12



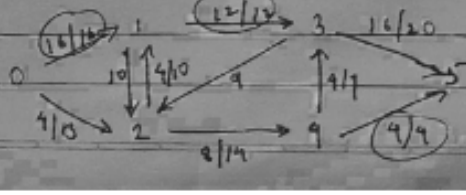
2. 0 → 2 → 4 → 5

min cap = 4



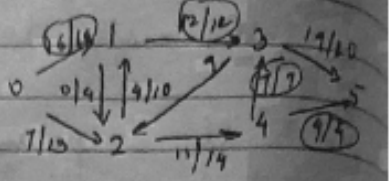
3. 0 → 1 → 2 → 4 → 3 → 5

min cap = 4



43159

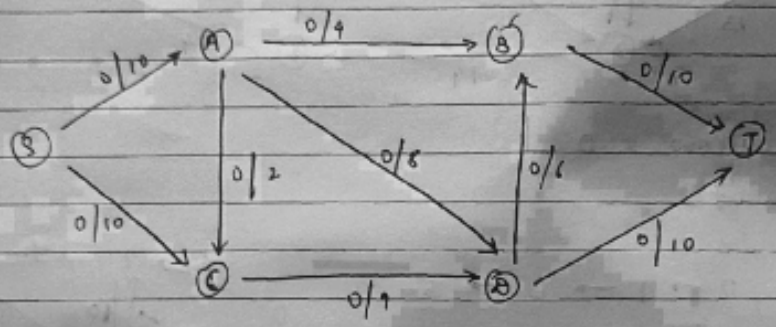
4. $0 \to 2 \to 4 \to 3 \to 5$



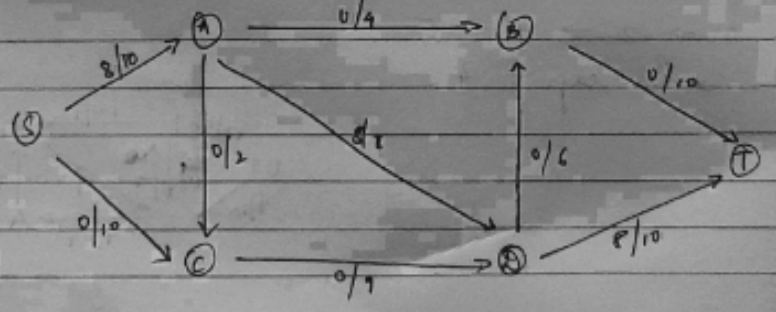
Now, since all incoming to ③ are filled & all the outgoing from ③ to ⑤ are filled, N/w has reached its maximum capacity.

∴ Total max. capacity = 23

Q.2.

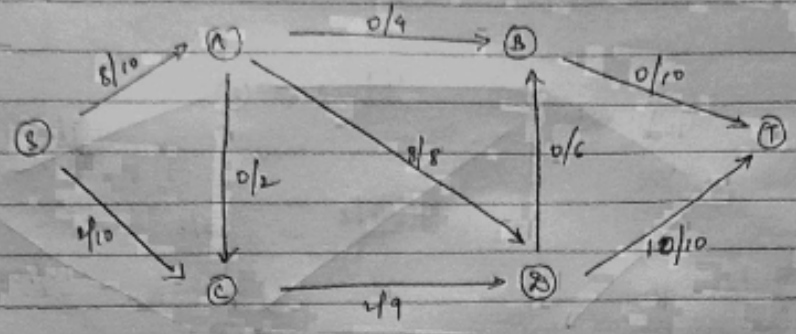


1. $S \to A \to D \to T = 8$

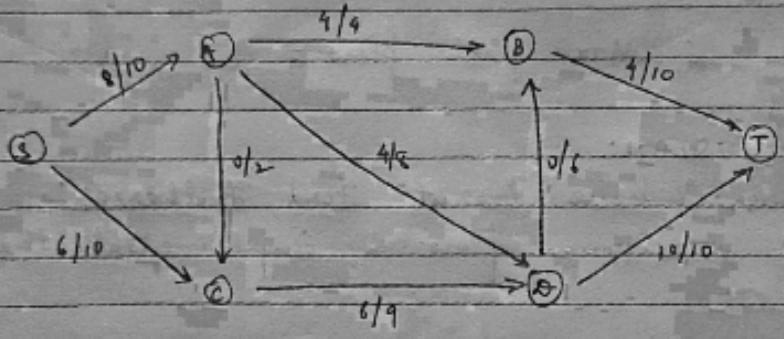


43159

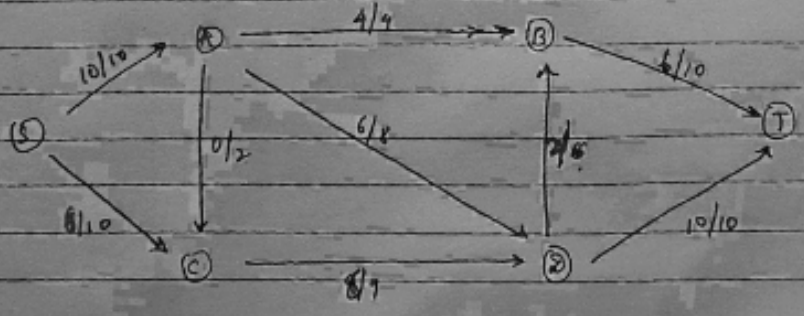
2. S → C → D → T = 2



3. S → C → D → A → B → T = 4



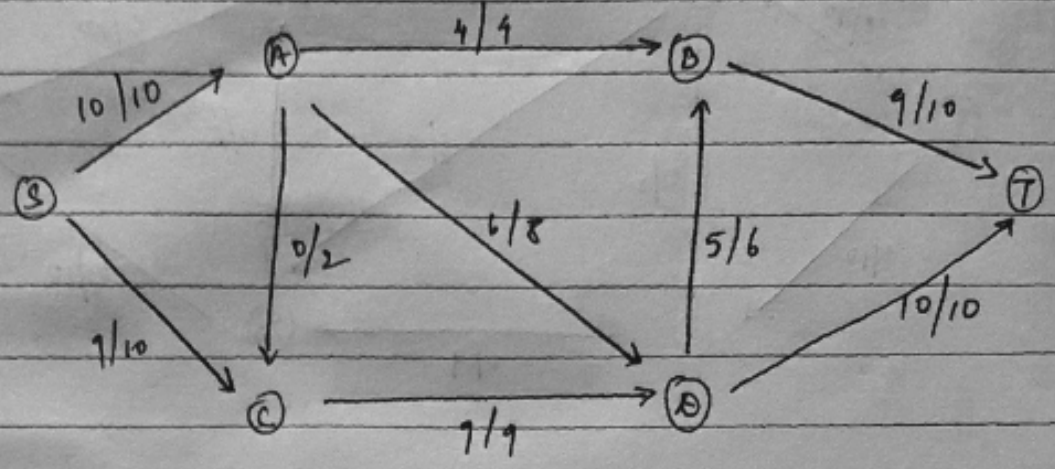
1. C → A → D → B → T = 2



43157

Page No.:
Date:

5. S → C → D → B → T = 3



All outgoing paths filled.

∴ Max path flow = 19

CONCLUSION:

Thus, I have learnt Ford Fulkerson's method for maximum flow & implemented the same.