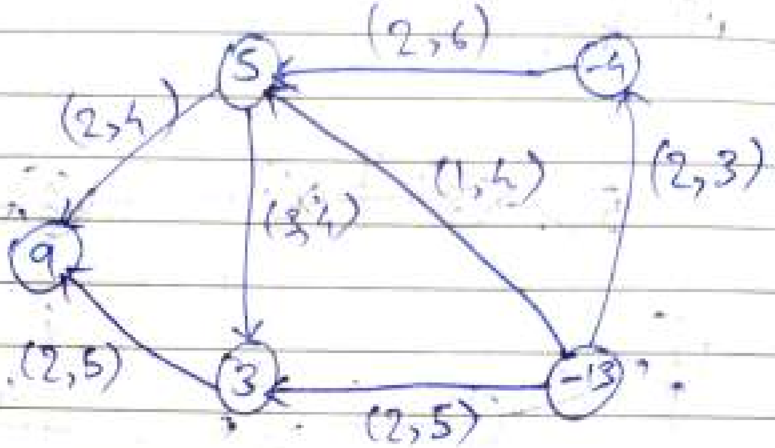
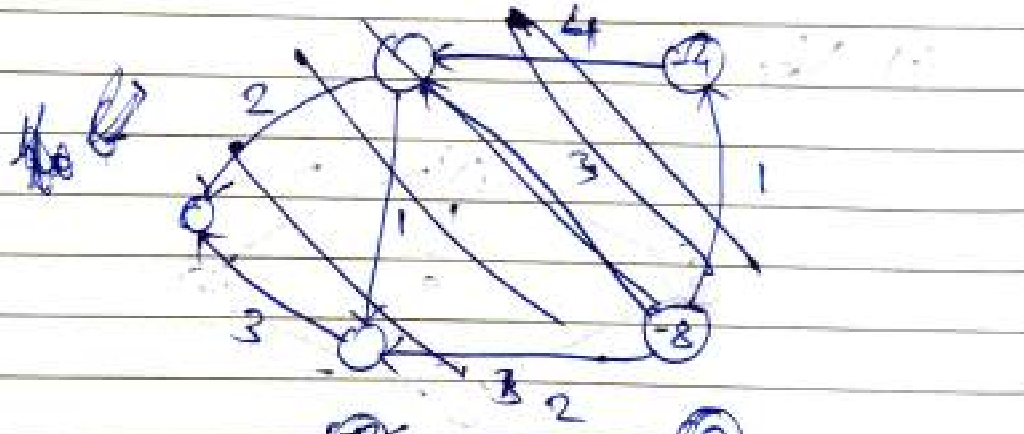
CSCI 570 #6498-8280-04

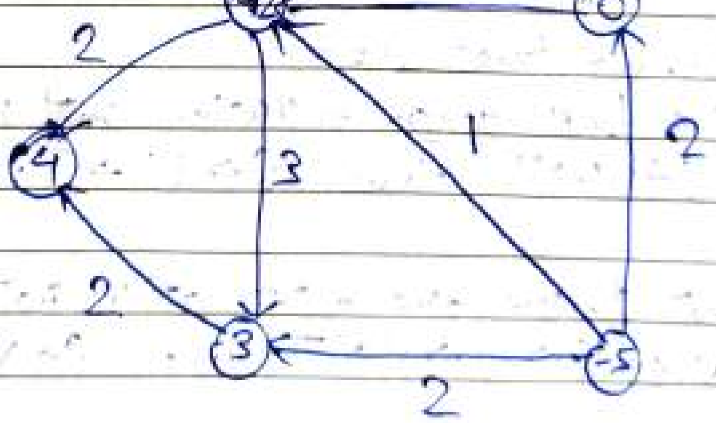
Homework #9

Q1

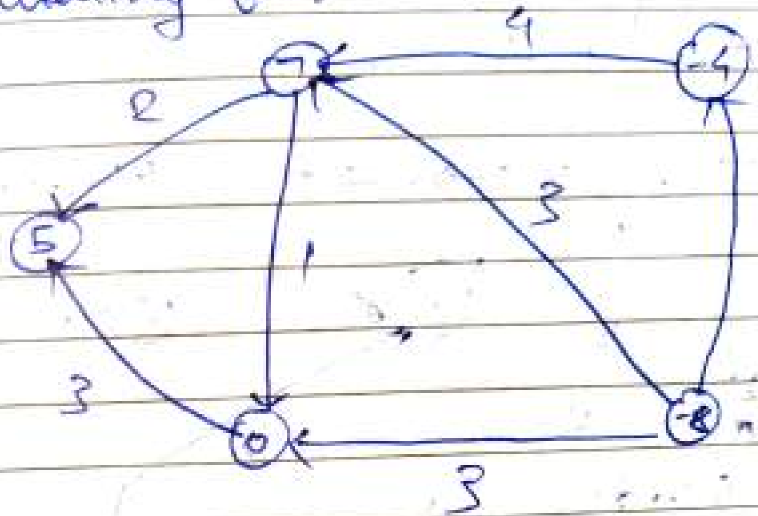


(a) After pushing the lower bound flow and calculating flow imbalance Lv:

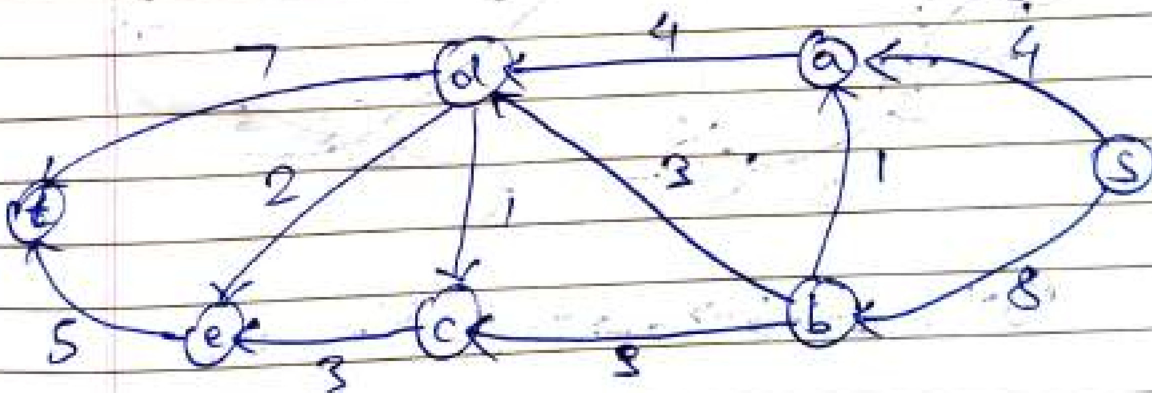




Calculating G':



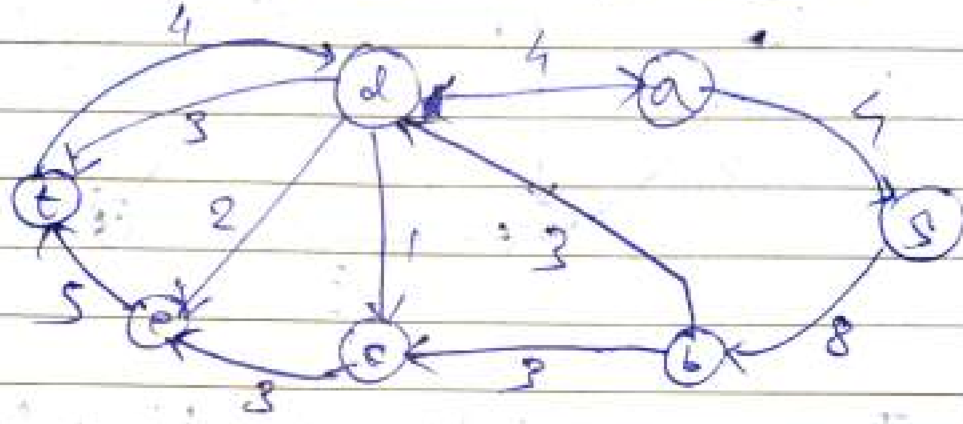
(b) To convert this to a Max Flow Problem, we will add a source S and sink T, while representing demands at all edges as vertices:



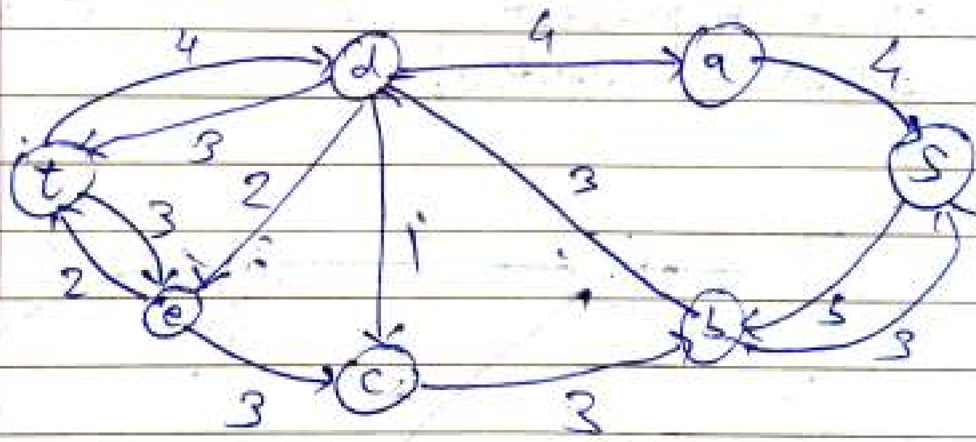
Max Flow Graph

(c) Solving the Max Flow problem using Ford Fulkerson:-

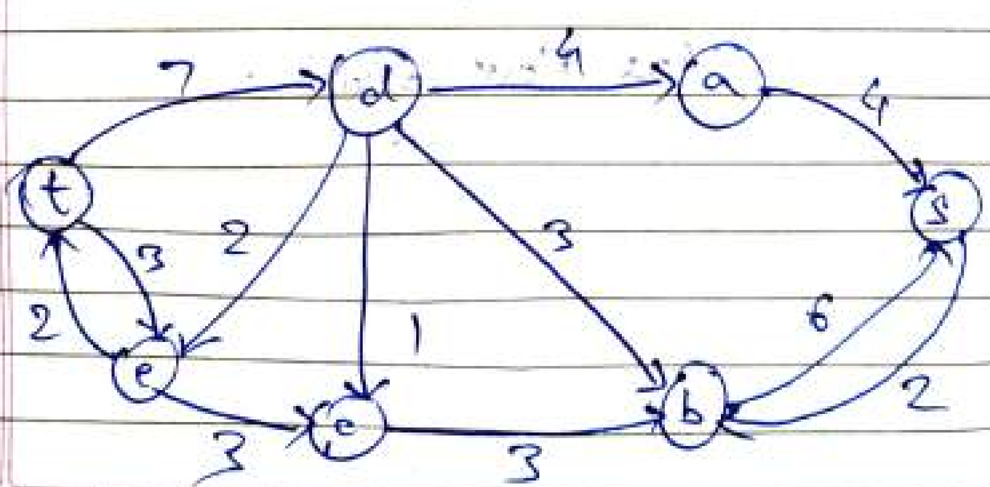
Path: s→a→d→t , Flow = 4



Path: s→b→c→e→t , Flow : 3



Path: s→b→d→t , Flow : 3



Here, total Max Flow: 10 and total demand value = 12.
Since $v(f) < \sum d_v$, there is no feasible circulation available for the problem.

Q2

The problem can be converted to a circulation network flow diagram. We create a graph G where:

(1) For every box i, we can create 2 nodes u_i and v_i. We connect these two nodes with an edge having a lower bound & capacity of 1 since a box cannot be reused.

(2) For each pair of boxes, if box j fits inside box i, an ~~node~~ edge will be drawn from node v_j to u_i of lower bound 0 and capacity of 1 because a box might not store the smaller box.

(3) We create a source node and a sink node to represent room space and empty boxes respectively, with demand values of $-k$ and k where k is the min. no. of boxes visible.

(4) For each box i, we create an edge from s to u_i and an edge from v_i to t with capacity 1 and lower bound 0.

<u>To Prove:</u> There is a possible arrangement with k visible boxes ~~and~~ if and only if there is a feasible circulation in G with demand $-k$ in the source and k in the sink.

<u>Proof:</u>

For each nesting of boxes, a path is generated from s to t. Hence, if there are k visible boxes, there are k paths. The path from s ~~to~~ will be the box that is visible, followed by those boxes which are nested inside of it. To find minimum k, we try finding all possible paths along with the constraints. This can be solved using a base network flow problem, which takes polynomial time.

Q3 The problem can be converted to a simple network flow problem. The following points are to be noted for Graph G:

(1) Create a node a_i for each family i and create a node b_j for each table j.

(2) To make sure that only one member of the family is allowed, we will connect a_i to b_j with a capacity of 1.

(3) Create a source node s and connect it to all family nodes with capacity g_i and a sink node t that connects with table vertices with a capacity h_j.

<u>To Prove:</u> There exists a valid seating solution if the max flow from source to sink is $g_1 + g_2 + \cdots + g_n$.

<u>Proof:</u> Here, ~~the~~ if ~~we~~ a member of family i sits on table j, we assign a flow for edge $(a_i, b_j) = 1$, else we assign 0. The capacity from ~~s~~ b_j to ~~a_i~~ t is h_j which is valid because it represents the number of seats that a table can accomodate at max. Also, the capacity from s to a_i is g_i is valid because it represents the number of members a family has.

Also, the values would be in integers since ~~form~~ it is trivial. Clearly, the assignment will be valid.

Conversely, assume that the value of max flow is $g_1 + g_2 + \cdots + g_n$. If we use FF, then edge weights are integers. If the max flow is $g_1 + g_2 + \cdots + g_n$, the edges from vertex s are saturated. Since every edge between family and table is either 0 or 1, each family is connected to g_i tables. Thus, we have a valid seating.

Q4

(a) The problem can be solved with max flow network. The graph G will have the following properties.

(1) The patients will be denoted by nodes p_i and hospitals will be denoted by node h_j. The edge capacity of $p_i \to h_j$ is 1 and we will connect to those hospitals which are within a half hour drive.

(2) Connect the patients to source s with edge capacities as 1 and connect hospitals h to sink t with edge capacities ~~n/(k+1)~~ $n/k + 1$.

We run FF algorithm to calculate the max flow

(b) Each path from s to t denotes one injured person going to a hospital which is a half-hour drive. The hospital also receives at most $n/k+1$ patients. We should have n patients in order to have a balanced allocation of all patients.

(c) The time complexity of Ford-Fulkerson is $O(Cm)$ where C is the max possible flow and m is the number of ~~nodes~~ edges.
Here, max flow $= C = n$ and ~~nodes~~ no. of edges $= n+k+nk$.

Thus, the time complexity is $O((n+k+nk)n)$
$= O(n^2k)$.