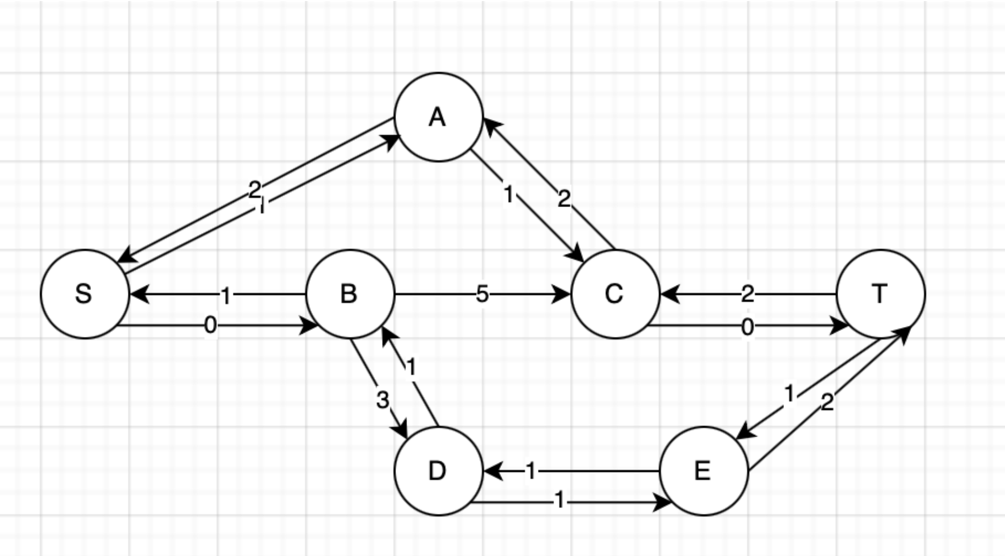
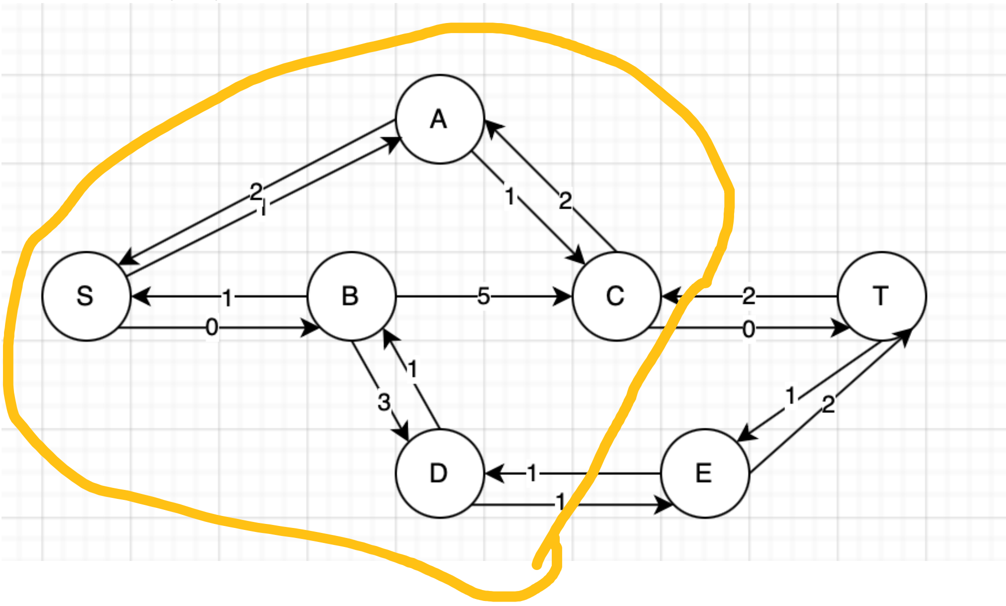
CSCI 570 - HW 4

1.
a.



b. max-flow=3
 min cut G1={S,A,B,C,D}
 G2={E,T}



min-cut = 3
min-cut==maxflow

2.
There are n traders t1,t2,…,tn
c1,c2,…,cn represents the currency and b1,b2,…,bn are available currency to trade Francs to respective currency

Fk represents the amount of Francs trader tk wants to trade
Skj is the amount of Francs which the trader tk is willing to trade for currency cj
So the source s will be connected with t1,t2,…,tn, and edge (s,ti) will have the capacity Fi
The edge (ti,bj) will have capacity Skj
The edge (bj,t) will have the capacity bj
Let f be the maximim flow in the network, now calculate the flow in the network
If the flow f is equal to the sum of francs all the traders want to trade then all the traders will be able to trade completely, i.e. f = (summation)k Fk

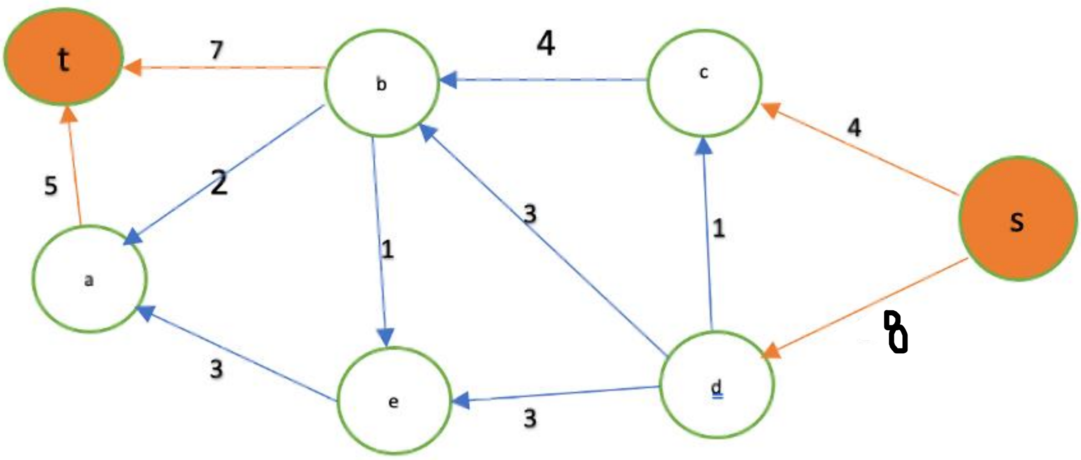
3.
There are n students s1,s2,…,sn
m is the maximum times a student can be selected
there are k in person-classes, c1,c2,…,ck
the source will be connected with c1,c2,…,ck and the edge (s,ci) will have the capacity 1
The edge (ci,si) will have the capacity 1
The edge (si,t) will have the capacity m
Now use the Ford-Fulkerson algorithm to find the maximum flow on this network
If the maximum flow in the network is equal to k, then the algorithm is said to be efficient

4.
a) Algorithm:
construct a network flow G' with all the edges and vertices from G and s and t as source and sink vertices, each edge has a max capacity of 1
connect the source vertex s with the starting vertices and t with the ending vertices
to find the maximum flow on this network we need to run the Ford-Fulkerson algorithm, and let the max flow be F
run the depth first search on the max flow and return all the possible paths, repeat this step k number of times since there are k vertices, if a edge is visited then mark it as visited and that edge cannot be used again
if the number of paths is equal to k, that is k number of paths then the algorithm returns true or else fail

b) in the above network flow add max capacity of every directed edge as 1, that means every vertex max in flow and max out flow is 1, so a vertex has been visited then the vertex reaches its max capacity and cannot take in any extra flow, and now if we run the above algorithm we will get the k different paths with no same edges and vertices

5.
a) compute $d'(v) = d(v)-(f^{in}(v)-f^{out}(v))$
$d'(a) = 9 – (4) = 5$
$d'(b) = 5 – (3-5) = 7$
$d'(c) = -4 – (2-2) = -4$
$d'(d) = -13 – (-5) = -8$
$d'(e) = 3 – (5-2) = 0$

b)



c)

In the original graph the sum of all the demand values is equal to zero, Therefore the feasible circulation exists