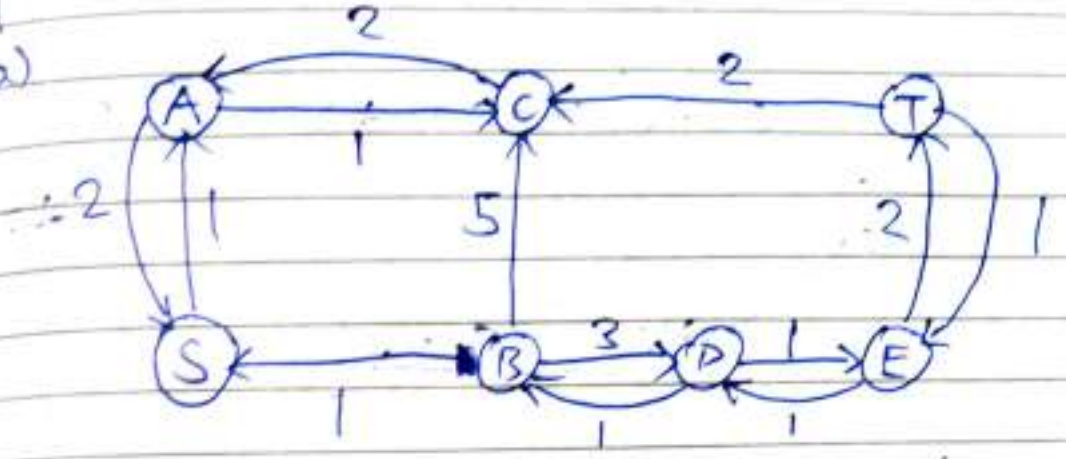
6498-8280-04 CSCI 570

Homework #8

Q1

(a)



Final Residual Graph

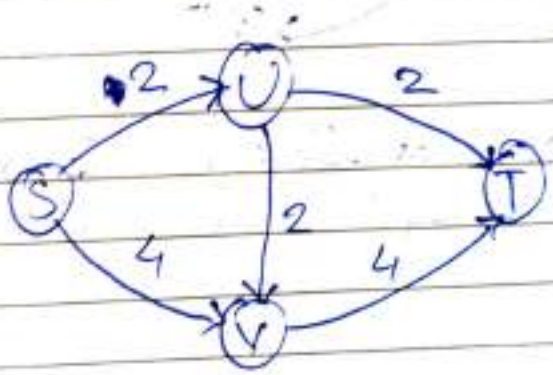
(b) The Max Flow is $1+2=3$

(c) The Min cut is $\{S, A, C\}$ and $\{B, D, E, T\}$

Q2

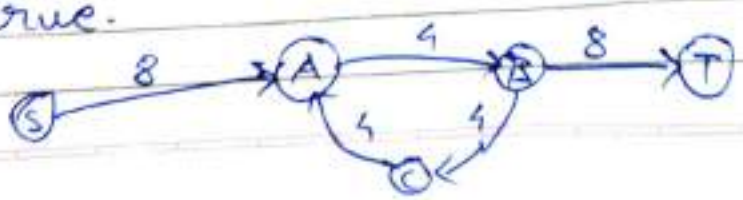
(a) True

Eg:



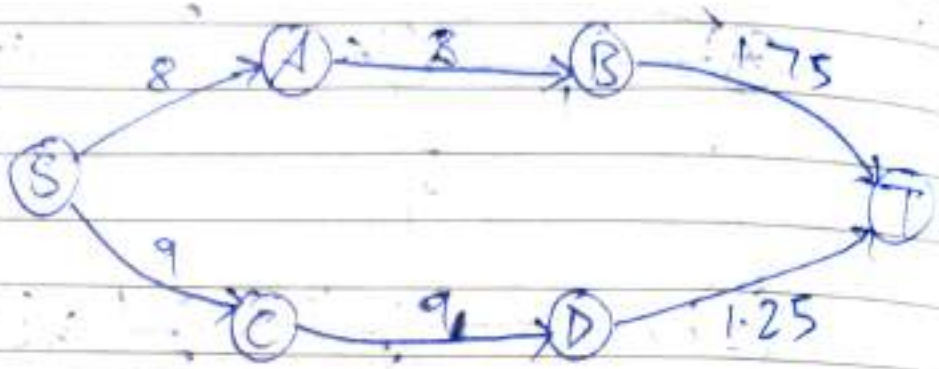
Here, max no. of edge disjoint paths = 2 and max flow = 6.

(b) True.



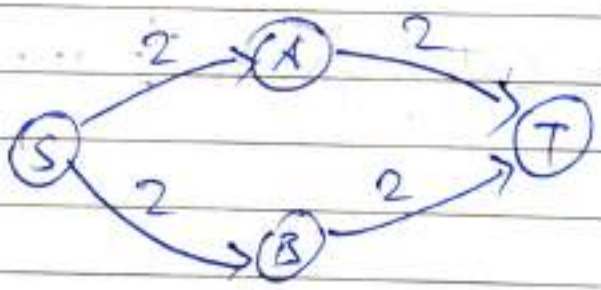
Here, max flow = 4

(c) False. Consider this graph:



Here, max flow = 1.75 + 1.25 = 3 which is an integer.

(d) False. Consider this graph:



Here, max flow is 4. If we increase all edges by 1, max flow is 6.

(e) True.

Even if we multiply all the edges by a positive number, the nodes that belong to a certain cut remain in that cut.

Q3

Let us assume g = initial max-flow of network. Since all the edges are unit capacity, so by decreasing k edges, the resulting max flow cannot be less than g-k.

If g is the max flow, it means that the min cut of the graph has g edges out of it.

So, if g<k, we remove all the edges from this min cut, reducing the max flow to 0. Else if g>k, we remove k edges from the min cut, reducing the max flow to g-k.

In any case, the max flow is always be equal to max(0, g-f).
With this, we get the maximum flow G' = (V, E-F)

Calculating the min cut takes linear time and removing k edges also takes linear time.

If the edges have more than unit capacity, the algorithm cannot guarantee the minimum possible max-flow.

Q4

We can solve this problem with the help of network flow.

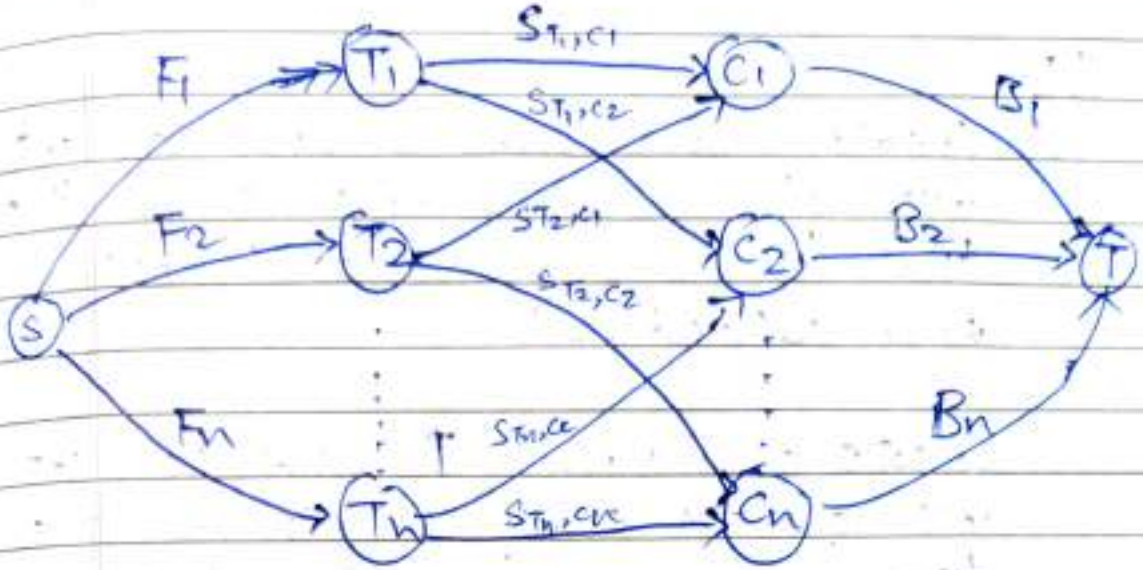
For this, we can have a source node 'S' which contains all the USD that the tourists need to convert. 'S' will be connected to each tourist, with edge capacity Fi which denotes the max. amount the tourist wants to convert.

On the other end, we will have a sink node 'T' which will be connected to all the currencies. The edge capacity would be Bi where it represents the max. amount of a currency.

Connect all tourists to currencies with the edge capacity $S_{i,j}$ where it represents the limit of Tourist i has to convert to a currency j

We will get a set of requests when we run the Ford-Fulkerson Max-Flow Algorithm on this network flow.

An example of the graph is given below



For forward direction, assuming all the assignments are valid, we will prove max-flow of graph = sum of all F.

Since all assignments are valid, it means that the bank can satisfy all requests for currency exchange i.e. sum of amount going from S = sum of all amount received at T.

For backward direction, assuming that we have a max-flow of sum of all F, it means all the edges from S are able to convert their currencies and the edge values are exhausted. This proves our assignment of values was valid.

Q5.

(a) Create a source node 's' and connect s with all the starting points for the tour $(a_1, a_2, \ldots, a_n) \in V$ with edge capacity = 1

Create a sink node 't' and connect t with all the ending points for the tour $(b_1, b_2, \ldots, b_n) \in V$ with edge capacity = 1.

Give all edge capacities = 1 from a_i to b_i.

If there are k no. of campus tours to be conducted, we will run any max flow algorithm and if max flow result = k, we can say that k campus tours can be conducted.

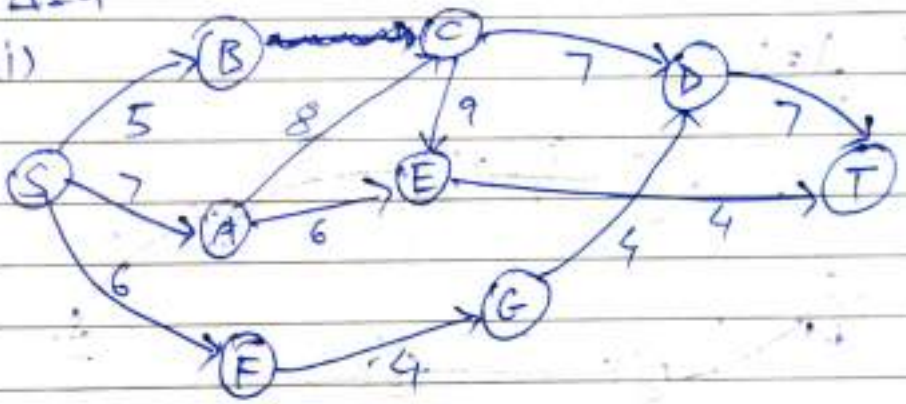
To ensure that a path is not traversed again, ~~when we run through a s-t path, we delete all the edges of that path~~ we have the edge capacity as 1.

(b) For this, we would split vertices V as V_1 and V_2, between an edge capacity 1. So, regardless of how many vertices come to V_1, there must be only one path that goes out from V_1, thus retaining the property that no vertices are shared.

Q6

$\Delta = 4$

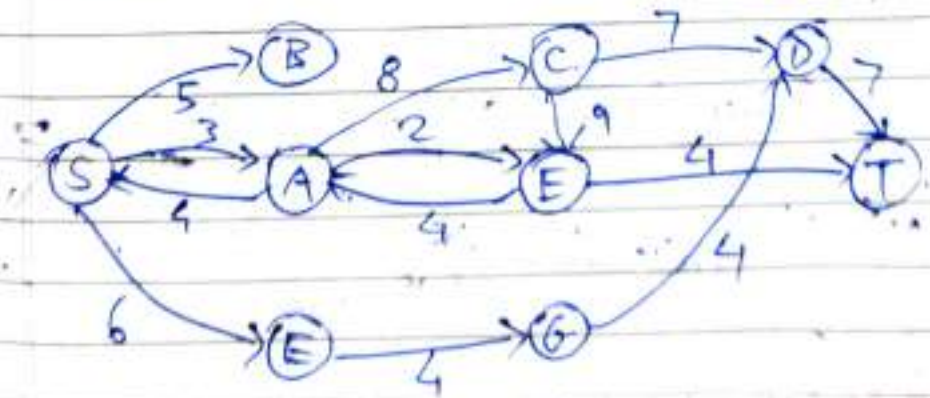
(a) (i)



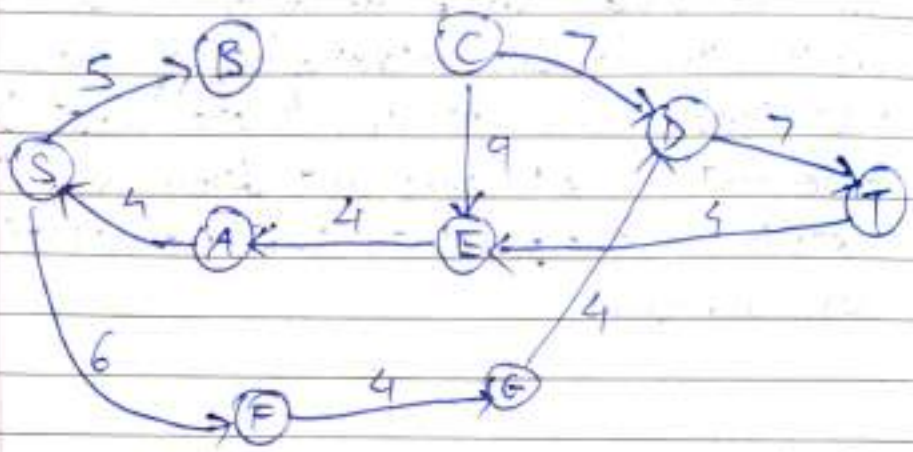
We choose the path $S \to A \to E \to T$

(ii)

Flow:

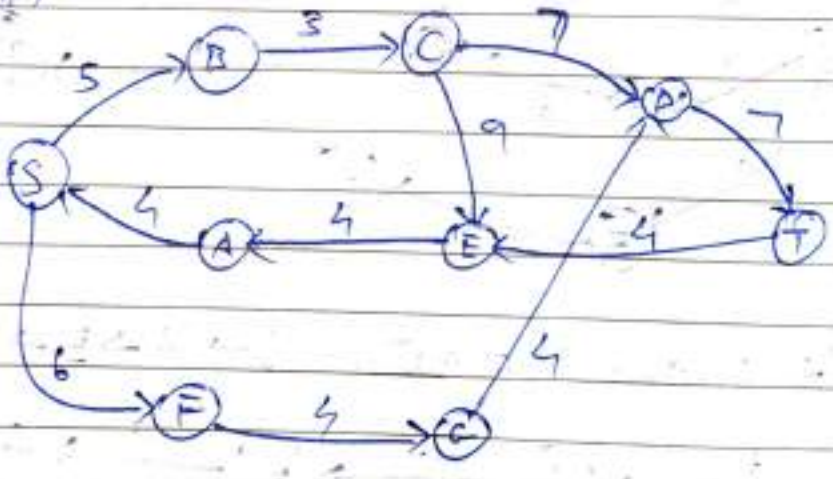


(iii) G_f =



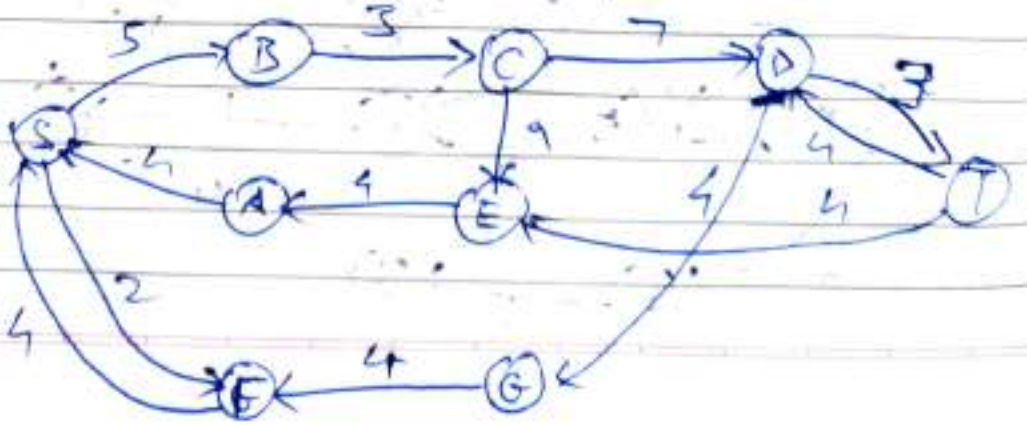
(b) $\Delta = 2$

(i)

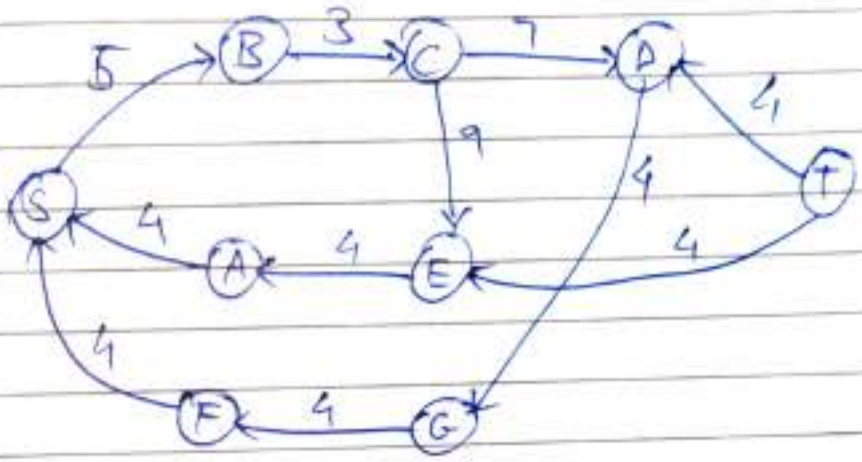


(ii) We select the path $S \to F \to G \to D \to T$

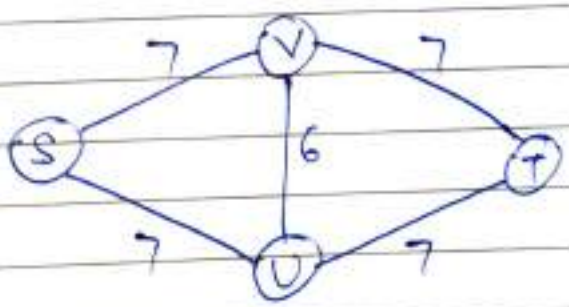
Flow:



(iii) G_f =



(c) Yes.



If we take $\Delta = 4$
The best augmentation paths are $S \to V \to T$ and $S \to U \to T$.

But there are alternative paths available are $S \to U \to V \to T$ and $S \to V \to U \to T$.

This increases the no. of iterations in the scaling phase.