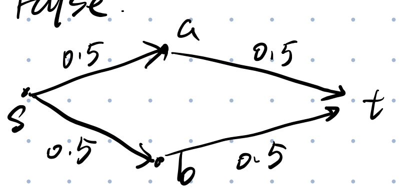
Q1

(a) True.

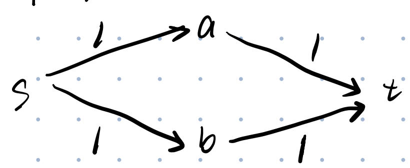
According to flow conservation, if there is a loop path with positive flow, then we can reduce the flow on this loop without violating the flow conservation properties of any node. This reduce does not affect the total flow from the source to the sink.

(b) False.



The max flow is 1.

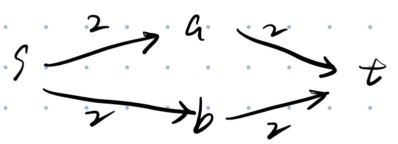
(c) False



The original max flow is 2

$f=2$

increase the capacity of every edge by 1.



The new max flow is 4.

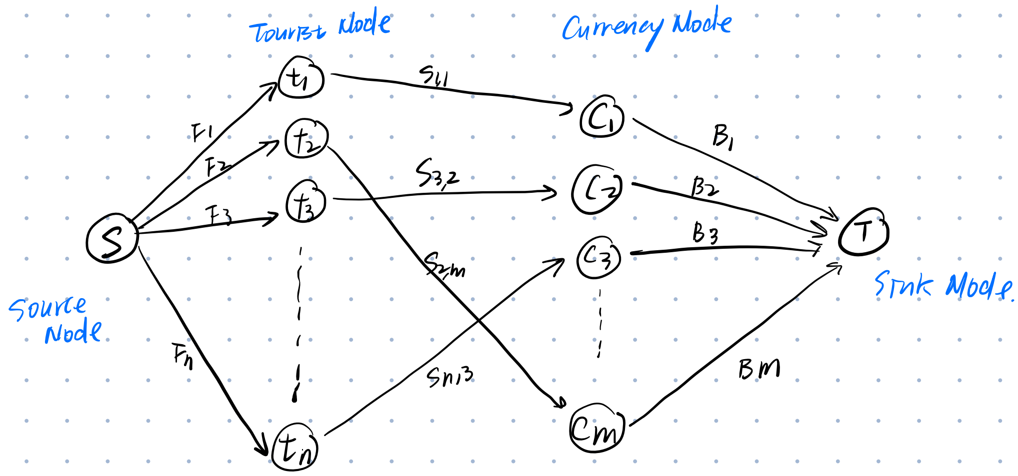
$f=4$

$\because$ $4 > 2+1$ $\therefore$ False

(d) True.

Multiplying all edge capacities by a positive scalar k doesn't change the relative capacities between the edges in the network. This means that if one path or cut had a smaller or larger capacity than another before scaling, it will maintain the relative property after scaling.

Q2



Run the network flow algorithm to find the maximum flow from s to T. If the maximum flow equals the sum of F_k for all tourists, it means all requests can be satisfied.

Proof:

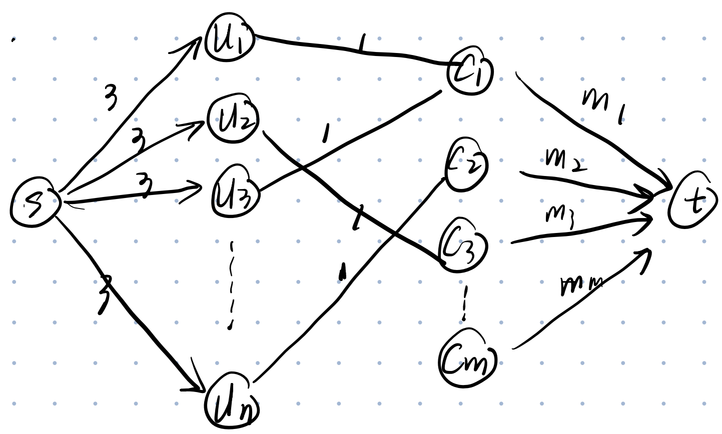
① If there is a way to satisfy all exchange requests, the maximum flow in the network will equal the sum of all USD to be converted.

Given the assumption, all edges from the source to each tourist node in the graph is fully utilized, which in turn means that the flow reaching the sink (total exchanged USD) is exactly $\sum F_k$, establishing the max-flow condition

② Conversely, if the maximum flow in the network equals the sum of all USD to be converted, then there is a way to satisfy all exchange requests.

Given the assumption, max flow $= \sum F_k$, means all capacities must be fully utilized. In other words, the flow through each of these edges equals their capacities. This indicates that each tourist can exchange the full amount they wish to exchange within the constraints provided by bank.

Q3.



Connect the source node to each user node u_i with an edge of capacity 3, since each user needs to connect to three towers.

For each user-tower pair, if the tower is within Δ distance, connect them with an edge of capacity 1.

Then, m_i is the maximum number of users that tower can support.

Use a polynomial-time algorithm to find the maximum flow in the graph. (Edmonds Karp)

Proof:

① If each user is connected to exactly three towers and no tower is connected to more than m_i users, then the max flow in the graph will be exactly $3n$.

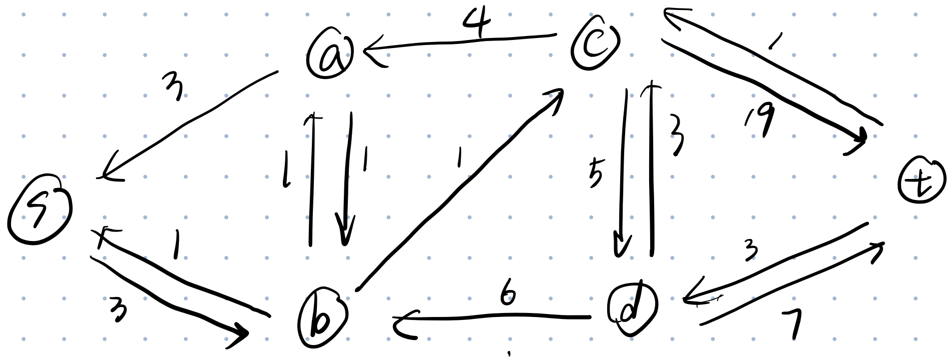
Given the assumption, all edges from the source to the user node is fully utilized. Since the graph is constructed to represent the constraints, and a valid assignment exists, the max flow must be equal to $3n$.

② Conversely, if the max flow in the graph is exactly $3n$, this implies that it was possible to assign each user to exactly three towers without exceeding the maximum capacity of each tower.

Assume that the max flow is $3n$. According to the max-flow min-cut theorem, this flow must fully use the capacities of the edges from s to each user node. Additionaly, since the flow is maximum, it must also respect the capacities of edges from the towers to T.

Q4

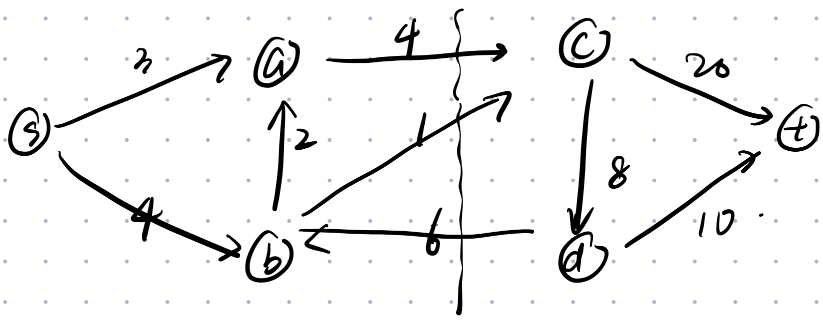
(a)



(b) Not max flow.

The max flow is 5.

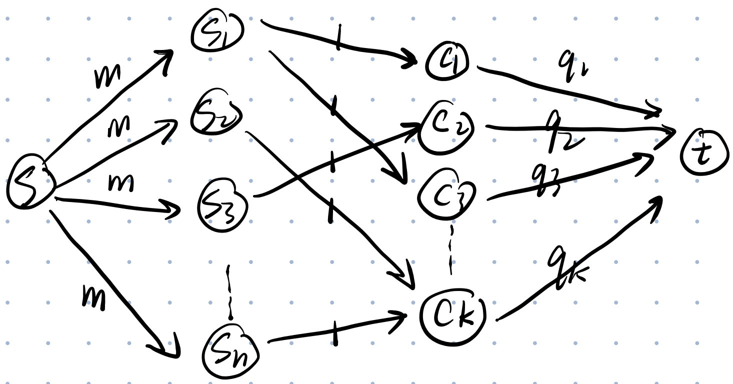
(c).



The min-cut is {s, a, b} and {c, d, t}.

Q5.

(a)



m: the minimum number of courses each student must enroll in to qualify as a full time student.

For each student s_j, connect them to the nodes of the courses in their desired group p_j with edge of capacity 1. This respresents the student's willingness to enroll in these courses.

q: the maximum number of students that each course can accommodate.

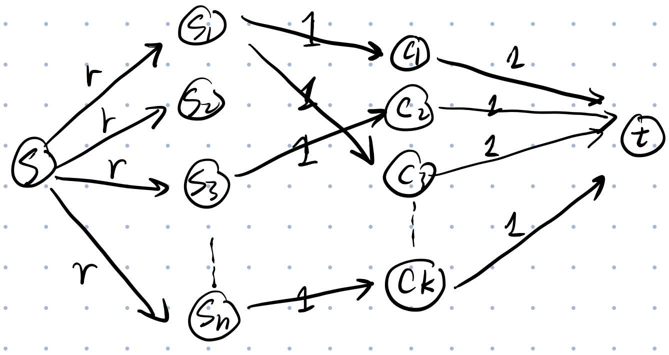
Use a polynomial-time flow algorithm (Edmonds-karp) to find the maximum flow from S to T.

Proof: A solution where every student can register as a full-time student exists if and only if the max flow from S to T equals n*m.

If every student can register in m courses, then for each student, there is a flow of m from S to the student node, and since every course can only acommodate up to q_i students, the capacity constraints for courses are not exceeded. Thus, the max flow must be n*m.

If the max flow is n*m, it means that each student node has a flow of m, and the courses have not exceeded their capacities q_i. Therefore, it's possible for each student to be enrolled as a full-time student.

(b)



Now, the C_i means student representative.

Each student can represent at most r courses => $S \xrightarrow{r} S_i$

Each course needs only 1 representative => $C_i \xrightarrow{1} t$

Connect S_i and C_j with an edge of capacity 1.

Use the same flow algorithm to find the maximum flow.

Proof: It's possible to appoint student representatives for each courses while adhere the conditions if and only if the max flow from S to T equals k (the number of course)

If there is a way to appoint representatives such that no student represents more than r courses, then the flow from each student node does not exceed r, and each course representative node will have a flow of 1 leading to the sink T. Since each course needs one representative, Thus max flow will be k.

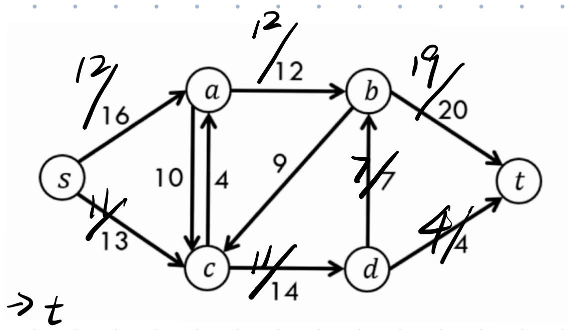
If the max flow is k, it implies that each course representative node has a flow of 1, indicating that each course has one representative and no student node has a flow exceeding r, thus adhering to the condition that a student could represent at most r courses.

Hence, it's possible to appoint the required representative under the given constraints.

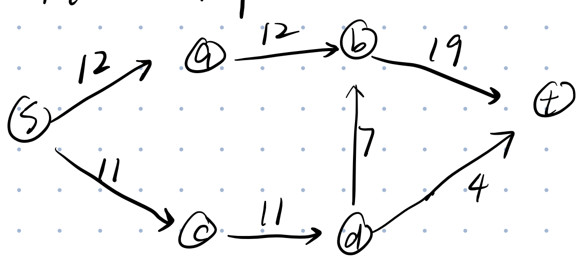
Q6.

(a)

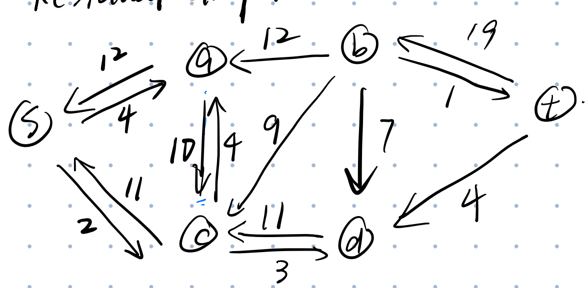
Iteration	Δ	path
1	16	No path
2	8	s → a → b → t
3	8	No path
4	4	s → c → d → b → t
5	4	s → a → c → d → t
6	4	s → c → a → s
7	4	No path
8	2	No path
9	1	No path



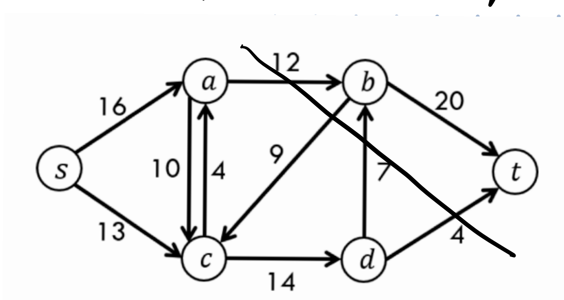
(b) Network Graph



Residual Graph

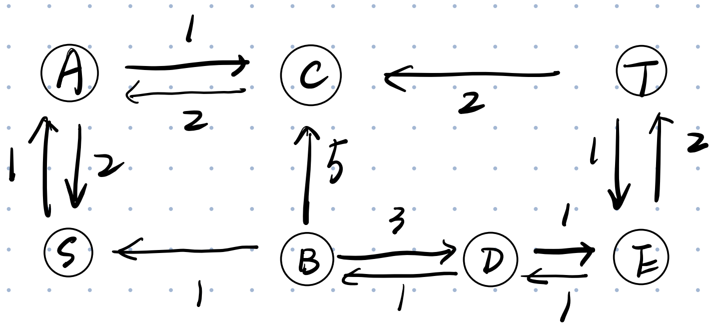


(c) Max flow is 23. Min-cut is {s, a, c, d} and {b, t}



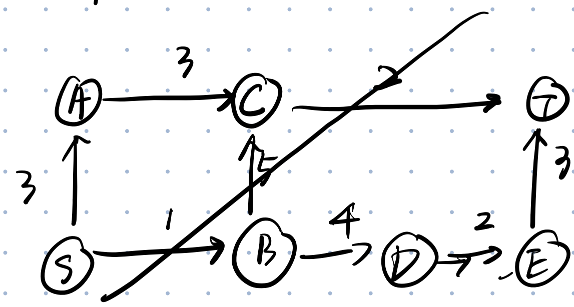
Q7

(a)



(b) Max flow is 3.

(c).



Mincut is {S, A, C} and {B, D, E, T}.

Q8

Let the current maximum flow in the network be g. After deleting k edges. the maximum flow is at least $g-k$. Because the capacity of each edge is 1. And Reduce 1 edge with at most reduce 1 in max flow. Moreover, the maximum flow from the source to the sink is equal to the capacity of any minimum cut the separates the source and sink.

The max flow can be either 0 or $g-k$. and it can not be decreased further due to the Max flow min cut theorem. And we will got the maximum reduction.

Case 1: Maxflow = 0.

if $g \le k$, delete all edges in min cut, reducing the max flow to 0. since the size of this minimum cut is less than or equal to k.

Case 2: Max flow = $g-k$

if $g > k$, delete k edges from the min cut, new max flow will be $g-k$.

Finding the minimum cut and deleting k edges can be done in polynomial time in terms of the size of network.

Futhermore, if the capacity is greater than 1. simply deleting k edges from the min cut can not guarantee achieving the minimum possible maximum flow. This is because in non-unit capacity scenarios, the importance of edges is no longer equal, and some edges may carry more flow than others