Student who wrote the solution to this question: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai
Students who read this solution to verify its clarity and correctness: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai

1. We can first construct a new directed graph $G'$ which has all the same vertices in G. And for each edge $(u, v)$ in $G$, adding not only $(u, v)$ but also $(v, u)$ to $G'$. Then we set the capacity of each edge $c(u, v)$ to 1, so that the minimum number of edges that must be removed to disconnect the graph will be equal to the minimum capacity of all $s$-$t$ cuts of the graph.

   Then we can pick an arbitrary vertex $s$ and run the maximum-flow algorithm on the flow network $\mathcal{F} = (G, s, t, c)$ for all vertices $t$ such that $t \neq s$. We claim that the minimum number of edges that must be removed to disconnect the graph is the minimum of those maximum flow value.

   There are $|V| - 1$ such flow network in total, each having $|V|$ vertices and $2|E|$ edges. So the maximum-flow algorithm will run on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges.

   *Proof.* Suppose for contradiction that the value our algorithm produce, call it $m$, is not the minimum number of edges that must be removed to disconnect the graph, and let $k$ denote the real minimum and $A, B$ denote the two resulting set of disconnected vertices. Then

$$k < m \tag{1}$$

   Let $f(s, t)$ denote the value of the maximum flow from vertex $s$ to $t$ in $G'$ where $s \in A$ and $t \in B$. By the max flow min cut theorem, $f(s, t)$ is a minimum cut and $k$ cannot be smaller than $f(s, t)$. Then

$$f(s, t) \leq k \tag{2}$$

   On the other hand, our algorithm produce the minimum of all maximum flows, so

$$m \leq f(s, t) \tag{3}$$

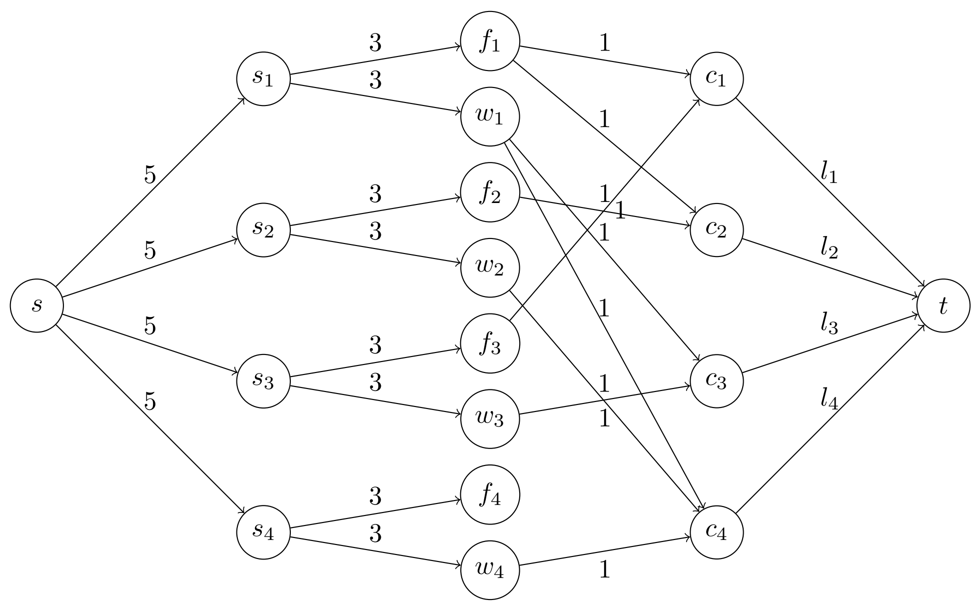
   (2) and (3) contradict with our assumption in (1). So our algorithm is correct.

Student who wrote the solution to this question: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai
Students who read this solution to verify its clarity and correctness: Zimo Li, Yu-Hsuan Chuang, Jiahong Zhai

2. (a) We can turn this problem into a network flow shown in the example below. In our example $S = \{s_1, s_2, s_3, s_4\}$, $C = \{c_1, c_2, c_3, c_4\}$, $T_1 = \{c_1, c_2\}$ and $T_2 = \{c_3, c_4\}$.

We introduced two vertices $f_j$ and $w_j$ for each student $s_j$ representing the bottleneck of the maximum number of courses they can take in each term and they have inflow capacity of 3. Each student $s_j$ has a inflow capacity of 5 from the source vertex $s$ that represents the total limit of courses that they can take. For $f_j$ and $w_j$, the edge $(f_j, c_i)$ and $(w_j, c_i)$ exists and has capacity 1 if and only if student $s_j$ is interested in course $c_i$. Edge $(c_i, t)$ exists and has capacity $l_i$.

The assignment that maximize the revenue can be found by finding an integral maximum flow of the network described above from $s$ to $t$. Our method is valid because the value of a flow in this network represents the number of enrolment in all courses in a possible assignment. The maximum flow also maximize the total enrolment and therefore maximize the revenue.



(b) **Variables:**

We use all variables already defined in the problem.

$a^1_{i\,j}$: student i take course j in term 1, then assign 1 student i not take course j in term 1, or course j is not in term 1, then assign 0.

$a^2_{i\,j}$: student i take course j in term 2, then assign 1 student i not take course j in term 2, or course j is not in term 2, then assign 0.

$y_i$: student i take 5 courses, then assign 1. student i not take 5 courses, then assign 0.

$b_{i\,j}$: course i has student j in it, then assign 1. Otherwise assign 0.

**Objective function:** max $y_1 + y_2 + ... + y_n$

**Constraints:**

$b_{1\,1} + b_{1\,2} + ... + b_{1\,n} <= l_1$

......

$b_{m\,1} + b_{m\,2} + ... + b_{m\,n} <= l_m$

$a^1_{1\,1} + a^1_{1\,2} + ... + a^1_{1\,m} <= 3$

......

$a^1_{n\,1} + a^1_{n\,2} + ... + a^1_{n\,m} <= 3$

$a^2_{1\,1} + a^2_{1\,2} + ... + a^2_{1\,m} <= 3$

......

$a^2_{n\,1} + a^2_{n\,2} + ... + a^2_{n\,m} <= 3$

$a^1_{1\,1} + a^1_{1\,2} + ... + a^1_{1\,m} + a^2_{1\,1} + a^2_{1\,2} + ... + a^2_{1\,m} <= 5$

......

$a^1_{n\,1} + a^1_{n\,2} + ... + a^1_{n\,m} + a^2_{n\,1} + a^2_{n\,2} + ... + a^2_{n\,m} <= 5$

**Justification:**

with variable $b_{i\,j}$, we can constraint each course has student less than enrolment limit.

with variable $a^1_{i\,j}$ and $a^2_{i\,j}$, we can constraint each student has less than 3 courses per term and less than 5 courses in 2 terms.

$y_i$ indicate whether a student has 5 courses. We can maximize all $y_i$ to reach our goal.