Homework 8

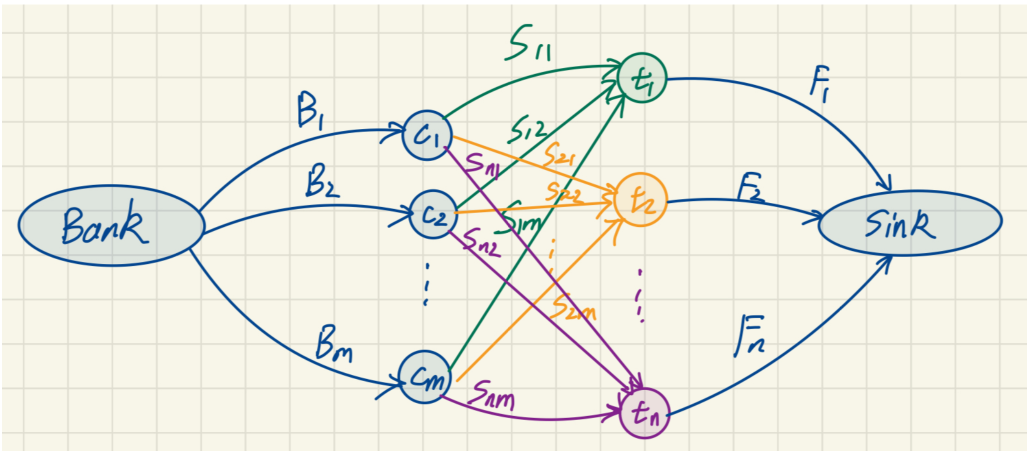
Jiahao Liu March 10, 2024

1. True or False. If true, give a brief explanation. If false give a counterexample.
(a) For a flow network, there always exists a maximum flow that doesn't include a cycle containing positive flow.
(b) If you have non-integer edge capacities, then you cannot have an integer max-flow value.
(c) Suppose the maximum s-t flow of a graph has value f. Now we increase the capacity of every edge by 1. Then the maximum s-t flow in this modified graph will have a value of at most f + 1.
(d) If all edge capacities are multiplied by a positive number k, then the min-cut remains unchanged.

(a) True. If there is a flow f with positive cycle C. We can just reduce the flow of each edge in C by $\min_{e \in C} f(e)$ to get f', obviously $v(f') = v(f)$ and there should be at least one edge in C its flow change to 0 in f', so we get a maximum flow f' without postivie flow cycle.
(b) False. Consider only two edges from s to t, both with capacity 0.5. The maximum flow 1 is an integer.
(c) False. Also consider only two edges from s to t, both with capacity 1, then v(f) = 2. However, the maximum flow after increasing each edge by 1 is 4 > v(f) + 1.
(d) True. The min-cut is the set of edges that are saturated in the maximum flow. If we multiply all edge capacities by k, the maximum flow will maintain with its value v(f) multiplied by k, so the min-cut remains unchanged.

2. A tourist group needs to convert all of their USD into various international currencies. There are n tourists $t_1, t_2, \ldots, t_n$ and m currencies $c_1, c_2, \ldots, c_m$. Each tourist t_k has F_k Dollars to convert. For each currency c_j, the bank can convert at most B_j Dollars to c_j.Tourist t_k is willing to trade at most S_{kj} of their Dollars for currency c_j. (For example, a tourist with 1000 dollars might be willing to convert up to 300 of their USD for Rupees, up to 500 of their USD for Japanese Yen, and up to 400 of their USD for Euros). Assume that all tourists give their requests to the bank at the same time. Design an algorithm that the bank can use to determine whether all requests can be satisfied. To do this, construct and draw a network flow graph, with appropriate source and sink nodes, and edge capacities. And Prove your algorithm is correct by making an if-and-only-if claim (10 points)

We need to construct a flow network with Bank as source, m edges from Bank to each Currency c_i with capacity B_i, n vertices as Tourist with edges from Currency c_j to Tourist t_k with capacity S_{kj}, and n edges from each Tourist to Sink with capacity F_k.

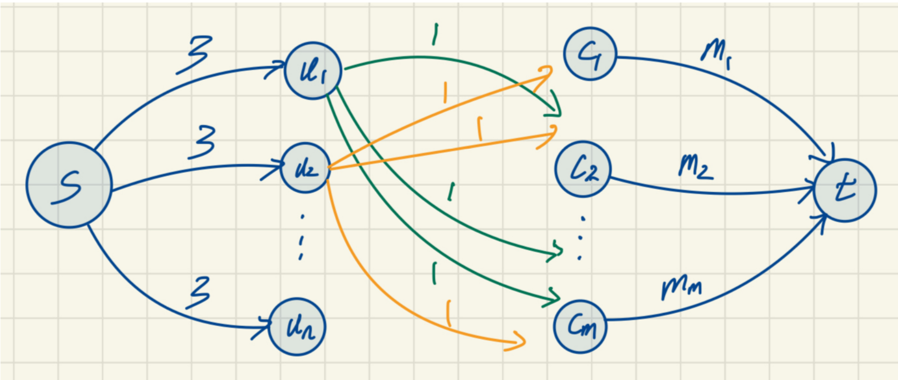


Then use Edmonds-Karp Algorithm to find the maximum flow of this Flow Network. If the maximum flow is equal to the sum of all tourists' Dollars, i.e. $v(f) = \sum_{i=1}^{n} F_i$, then all requests can be satisfied. Otherwise, not all requests can be satisfied.

Proof. Because the value of any flow f in above network is as most the sum of edge capacities into Sink, i.e. $v(f) \le \sum_{i=1}^{n} F_i$, so our maximum flow f^* must follow $v(f^*) \le \sum_{i=1}^{n} F_i$. So all requests can be satisfied $\iff$ there is a flow in this network with value equals $\sum_{i=1}^{n} F_i$ $\iff$ the value of maximum flow of this network is $\sum_{i=1}^{n} F_i$
If $v(f^*) = \sum_{i=1}^{n} F_i$, then obviously all requests can be satisfied.
If all requests can not be satisfied, then there must be at least one tourist t_k with $v(e_{t_k\ to\ Sink}) < F_k$, so the value of all possible flow must follow $v(f) < \sum_{i=1}^{n} F_i$, i.e. the value of maximum flow less than $\sum_{i=1}^{n} F_i$. □

3. You are given a collection of n points $U = \{u_1, \ldots, u_n\}$ in the plane, each of which is the location of a cell-phone user. You are also given the locations of m cell-phone towers, $C = \{c_1, \ldots, c_m\}$. A cell-phone user can connect to a tower if it is within distance Δ of the tower. For the sake of fault-tolerance each cell-phone user must be connected to at least three different towers. For each tower c_i you are given the maximum number of users, m_i that can connect to this tower. Give a polynomial time algorithm, which determines whether it is possible to assign all the cell-phone users to towers, subject to these constraints. Prove your algorithm is correct by making an if-and-only-if claim. (You may assume you have a function that returns the distance between any two points in O(1) time.)

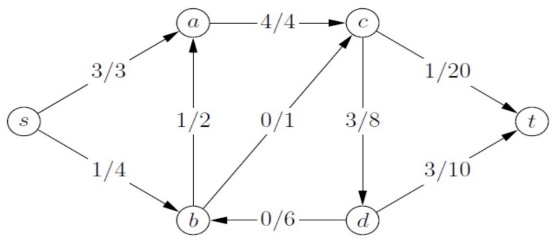
We need first check if any user u_i is in the Δ distance of tower c_j, then construct a flow network with a Virtual source and sink, n edges from source to each user u_i with capacity 3, m edges from each tower to Virtual sink with capacity m_i, and edges from each user to some towers with capacity 1 iff this user is in distance Δ of these towers.

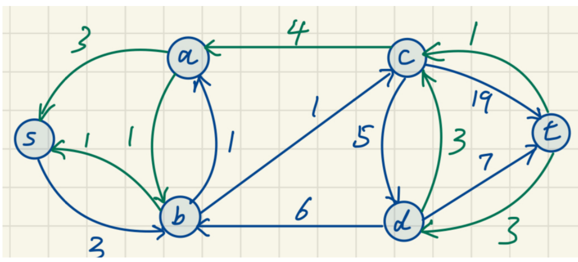


Then use Edmonds-Karp Algorithm to find the maximum flow of this Flow Network. If the maximum flow is equal to 3n, then it is possible to assign all the cell-phone users to towers, subject to these constraints. Otherwise, it is not possible.

Proof. Because the value of any flow f in above network is as most the sum of edge capacities from Source, i.e. $v(f) \leq 3 * n$, so our maximum flow f^* must follow $v(f^*) \leq 3n$. So all constraints can be satisfied $\iff$ there is a flow with value 3n in the above network $\iff$ the maximum flow in the above network is of value 3n.
If $v(f^*) = 3n$, then obviously all constraints can be satisfied.
If all constraints can not be satisfied, then either $\sum_{j=1}^{m} m_j < 3n$ or there is at least one user u_i with $v(e_{S\ to\ u_i}) < 3$, so the value of all possible flow must follow $v(f) < 3n$, i.e. the value of maximum flow less than $3n$. $\square$

4. You are given a directed graph which after a few iterations of Ford-Fulkerson has the following flow. The labeling of edges indicate flow/capacity.
(a) Draw the corresponding residual graph.
(b) Is this a max flow? If yes, indicate why. If no, find max flow.
(c) What is the min-cut



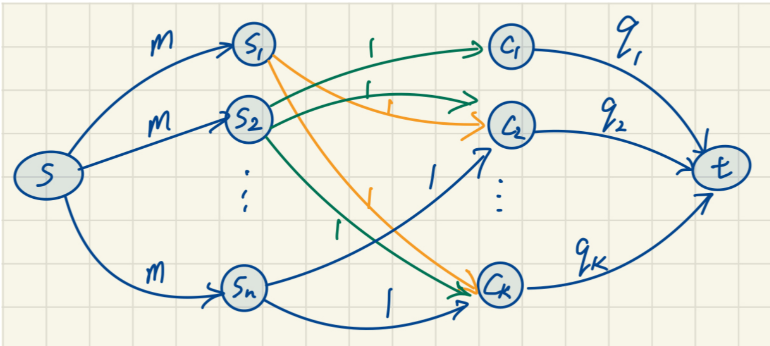


(a) The corresponding residual graph is as above.
(b) No, there is stil a simple s-t path s-b-c-t with value 1, so the maximum flow is 5.
(c) The min-cut is $(\{s, a, b\}, \{c, d, t\})$ with capacity 5.

5. USC has resumed in-person education after a one-year break, with k on-site courses available this term, labeled c_1 through c_k. Additionally, there are n students, labeled s_1 to s_n, attending these k courses. It's possible for a student to attend multiple on-site courses, and each course will have a variety of students enrolled.

(a) Each student s_j wishes to enroll in a specific group p_j of the k available courses, with the condition that each must enroll in at least m courses to qualify as a full-time student (where p_j is greater than or equal to m). Furthermore, every course c_i can only accommodate a maximum of q_i students. The task for the school's administration is to verify whether every student can register as a full-time student under these conditions. Propose an algorithm to assess this scenario. Prove your algorithm is correct by making an if-and-only-if claim.

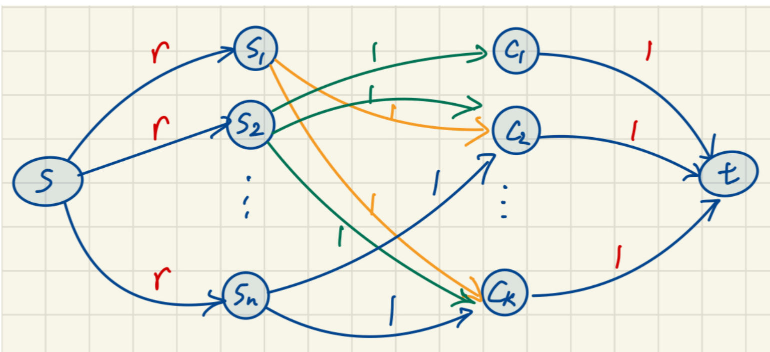
Similar to Q3, we can construct a flow network with a Virtual source, n edges from source to each student s_i with capacity m, k edges from each course c_j to Virtual sink with capacity q_j, and edges from each student to some courses with capacity 1 iff this course is in his specific group p_j.



Then use Edmonds-Karp Algorithm to find the maximum flow of this Flow Network. If the maximum flow is equal to mn, then it is possible to register all students under these conditions. Otherwise, it is not possible.

Proof. Because the value of any flow f in above network is as most the sum of edge capacities from Source, i.e. $v(f) \leq m * n$, so our maximum flow f^* must follow $v(f^*) \leq mn$. So all students can be registered $\iff$ there is a flow with value mn in the above network $\iff$ the maximum flow in the above network is of value mn.
If $v(f^*) = mn$, then obviously all students can be registered.
If all students can not be registered, then either $\sum_{j=1}^{k} q_j < mn$ or there is at least one student s_i with $v(e_{S\ to\ s_i}) < m$, so the value of all possible flow must follow $v(f) < mn$, i.e. the value of maximum flow less than mn. □

(b) Assuming a viable solution is found for part (a) where each student is enrolled in exactly m courses, there arises a need for a student representative for each course from among the enrolled students. However, any single student can represent at most r (where r is less than m) courses in which they are enrolled. Develop an algorithm to check whether it is possible to appoint such representatives, building on the solution from part (a) as a foundation. Prove your algorithm is correct by making an if-and-only-if claim.
Consturct a flow network similar to (a) except that the capacity of edges from Virtual source to each student is r (not m) and the edges capacity from each course c_j to Virtual sink is 1 (not q_j). And the course selections (edges between s_i to c_j) for each student will be reduced to what is available in the Max Flow from part (a). i.e. Remove some edges between students and courses that they did not get to sign up for.

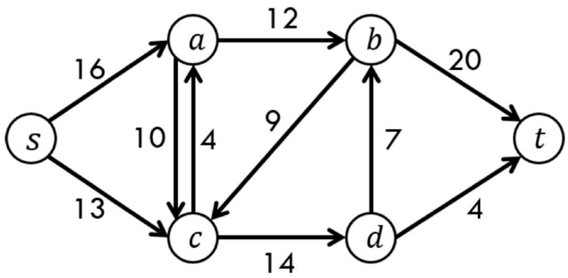


Then use Edmonds-Karp Algorithm to find the maximum flow of above Flow Network. If the maximum flow is equal to k, then it is possible to appoint such representatives. Otherwise, it is not possible.

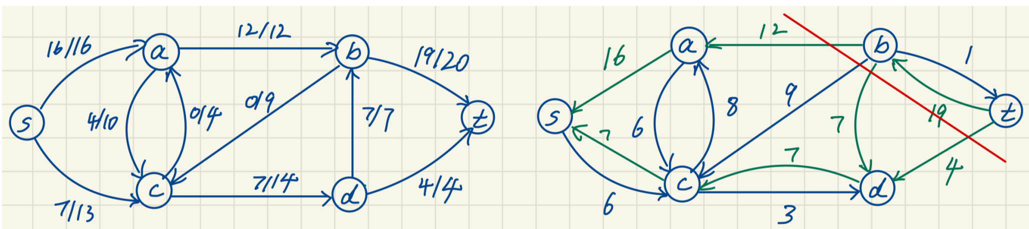
Proof. Because the value of any flow f in above network is as most the sum of edge capacities into Sink, i.e. $v(f) \leq k$, so our maximum flow f^* must follow $v(f^*) \leq k$. So the representatives can be found $\iff$ there is a

flow with value k in the above network $\iff$ the maximum flow in the above network is of value k.
If $v(f^*) = k$, then obviously can find representatives for all courses.
If not all courses can not be represented, then there is at least one student c_j with $v(e_{c_j\ to\ t}) = 0$, so the value of all possible flow must follow $v(f) < k$, i.e. the value of maximum flow less than k. $\square$

6. Given the below graph solve the below questions using scaled version of Ford-Fulkerson.
(a) Give the Δ and path selected at each iteration.
(b) Draw the final network graph and the residual graph.
(c) Find the maxflow and mincut.

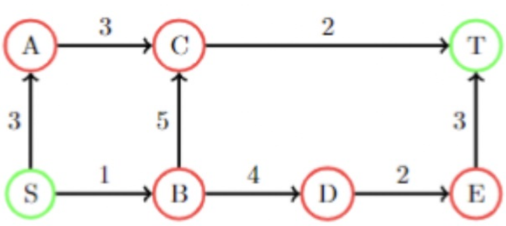


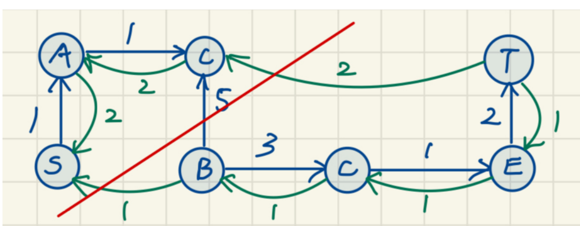
(a) When $\Delta = 16$, no simple path from s to t.
When $\Delta = 8$, choose path s-a-b-t with 12 flow.
When $\Delta = 4$, choose path s-a-c-d-t with 4 flow, path s-c-d-b-t with 7 flow.
When $\Delta = 2$, no simple path from s to t.
When $\Delta = 1$, no simple path from s to t.
(b) The final flow and residual graph are as follows:



(c) The maximum flow value is 12 + 7 + 4 = 23, the min-cut is $(\{s, a, c, d\}, \{b, t\})$ with capacity 23.

7. The following graph G has labeled nodes and edges between them. Each edge is labeled with its capacity.
(a) Draw the final residual graph G_f using the Ford-Fulkerson algorithm corresponding to the max flow. Please do NOT show all intermediate steps.
(b) What is the max-flow value?
(c) What is the min-cut?

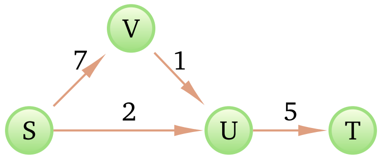




(a) As shown in the above figure.
(b) The maximum flow value is 2 + 1 = 3.
(c) The min-cut is $(\{S, A, C\}, \{B, C, E, T\})$ with capacity 3.

8. You are provided with a flow network where each edge has a capacity of one. This network is represented by a directed graph G = (V, E), including a source node s and a target node t. Additionally, you are given a positive integer k. The objective is to remove k edges to achieve the greatest possible reduction in the maximum flow from s to t in G. Your task is to identify a subset of edges F within E, where the size of F equals k, and removing these edges from G results in a new graph G' = (V, E-F) where the maximum flow from s to t is minimized. Propose a polynomial-time strategy to address this issue.

First find the maximum flow in the original graph G using Edmonds-Karp Algorithm, then use the residual graph G_f to get the min-cut (A, B) of the original graph G. Consider the set of edges F' = $\{(u, v)|u \in A, v \in B\}$, then:
If the size of F' equals k, let F = F';
If the size of F' is less than k, then add any other k-|F'| edges in E-F' to get F.
If the size of F' is greater than k, then take any k edges in F' to get F.
Because we prioritize removing edges from the min-cut (A, B), the min-cut of the new graph G' should still be (A, B). Now that we have minimized the value of this cut, the value of maximum flow should also be minimized.
Furthermore, consider if the capacities of the edges are greater than one, and discuss whether your strategy still ensures the lowest possible maximum flow.
According to the Discussion Q4 during lecture, we know the "most vital edge" has no direct relationship with the min-cut, so the strategy DOES NOT work anymore. The following is just a counterexample.



Where the min-cut is $(\{S, V\}, \{U, T\})$ with capacity 3, and the maximum flow is 3. If k = 1, we should remove edge (U, T) rather than the edge in the min-cut.