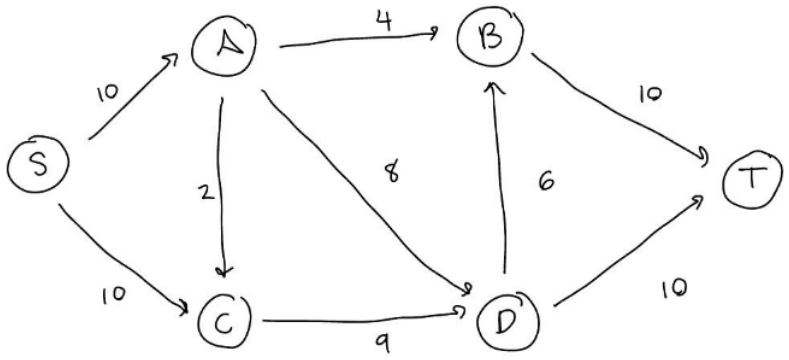
1: Title 1

Define the Following

- Flow Network:
- Maximum Flow Problem:
- Min-Cut Problem:
- Ford-Fulkerson Algorithm:
- Min-Cut Max-Flow Theorem:

1. Find the maximum flow of the following graph:



2. You are given a bipartite graph, where the left nodes are students, and the right nodes are employers. There are only edges between students and employers who would like to work with each other. Design a poly-time algorithm that will be able to assign students to employers such that there is a maximum matching (assign as many student-employer pairs as possible).

3. You are a CMU advisor, and have n students trying to sign up for classes. Each student can take a maximum of m classes. There are k classes available for sign-up, which can each take c_i students, for each class i. Design an algorithm that will return an assignment of the students such that each student is signed up for as many classes as possible.

4. What if students don't have the prerequisites for certain classes? Each student has a list of classes $1, 2.....c$ that they are able or willing to take. how can you alter the answer to the previous question to implement this new parameter?

5. Suppose you are given a directed graph $G = (V, E)$, with a positive integer capacity c_e on each edge e, a designated source $s \in V$, and a designated sink $t \in V$. You are also given an integer maximum s-t flow in G, defined by a flow value f_e on each edge e. Now suppose we pick a specific edge $e \in E$ and increase its capacity by one unit.

   (a) Prove that the maximum flow value can go up by at most 1.

(b) Show how to find a maximum flow in the resulting capacitated graph in time $O(m+n)$, where m is the number of edges in G and n is the number of nodes.

Recommended Video:

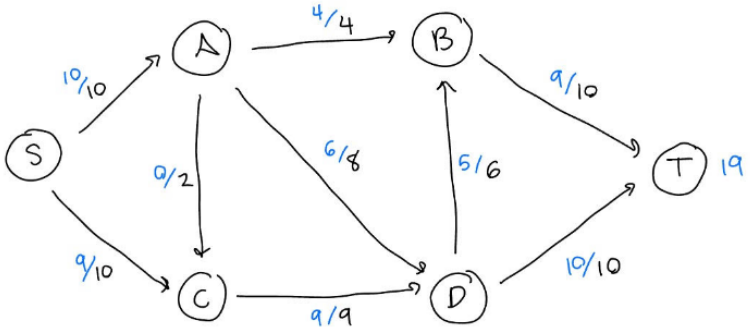
- Ford-Fulkerson in 5 minutes, Micheal Sambol, `https://youtu.be/Tl90tNtKvxs?si=1FPRBdobpX1d1eEM`

2: Solutions

Define the Following

- Flow Network: a connected, directed graph, where each edge contains a capacity of maximum flow, and a source vertex (s), and sink vertex (t)
- Maximum Flow Problem: Given a flow network G, find a flow f to maximize $v(f)$ (The amount of flow that travels from s to t
- Min-Cut Problem: Find a cut with the lowest cost to disconnect s from t
- Ford-Fulkerson Algorithm: Algorithm that allows us to find the max flow of a graph in O(maxFlow*E) time.
- Min-Cut Max-Flow Theorem: Let the graph have a max flow of $v(f)$, then there exists some cut in G that has a capacity of $v(f)$. The maximum flow in the graph is equal to the capacity of the minimum cut.

1. Find the maximum flow of the following graph: (See video linked below for how to get the answer)



2. You are given a bipartite graph, where the left nodes are students, and the right nodes are employers. There are only edges between students and employers who would like to work with each other. Design a poly-time algorithm that will be able to assign students to employers such that there is a maximum matching (assign as many student-employer pairs as possible).

   (a) To solve a bipartite matching problem, we will use Ford-Fulkerson. We will alter the bipartite graph as such: create a source node s that connects to all of the students. Do the same from the employers to a sink node t. Assign all edges to have a weight of 1 (a student can only match with one employer). Use Ford-Fulkerson to find the maximum flow. Think about why the maximum flow in the bipartite graph will also be the maximum matching. This algorithm will run in at most O(M*E), where M is the number of matches in the maximum matching.

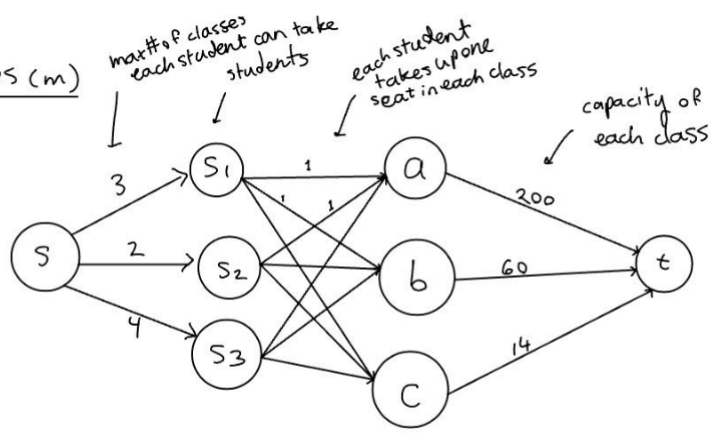
3. You are a CMU advisor, and have n students trying to sign up for classes. Each student can take a maximum of m classes. There are k classes available for sign-up, which can each take c_i students, for each class i. Design an algorithm that will return an assignment of the students such that each student is signed up for as many classes as possible.

(a) First, turn this problem into a directed graph, such that it can be solved using Ford-Fulkerson. Let there be a source s, and a line of nodes representing each student, and each class. For a line from s to the first student, have that edge have a capacity of m, the maximum number of classes they can take. Connect the student to every class, with each edge having a capacity of 1. This is because the student will only be able to fill one seat in each class. Then, connect each class to the sink node t, with each edge having the capacity x_i, for the corresponding class. Below is roughly what this graph would look like. Then use Ford-Fulkerson to find the maximum flow from s to t, which allows us to fill each classroom as much as possible.

Example

Student	max # of classes (m)
S 1	3
S 2	2
S 3	4

class	capacity
a	200
b	60
c	14



4. What if students don't have the prerequisites for certain classes? Each student has a list of classes $1, 2.....c$ that they are able or willing to take. how can you alter the answer to the previous question to implement this new parameter?

   (a) Instead of connecting each student to every class, only connect those students to the classes they can take. Below is an altered version of the graph above.

5. Suppose you are given a directed graph $G = (V, E)$, with a positive integer capacity c_e on each edge e, a designated source $s \in V$, and a designated sink $t \in V$. You are also given an integer maximum s-t flow in G, defined by a flow value f_e on each edge e. Now suppose we pick a specific edge $e \in E$ and increase its capacity by one unit.

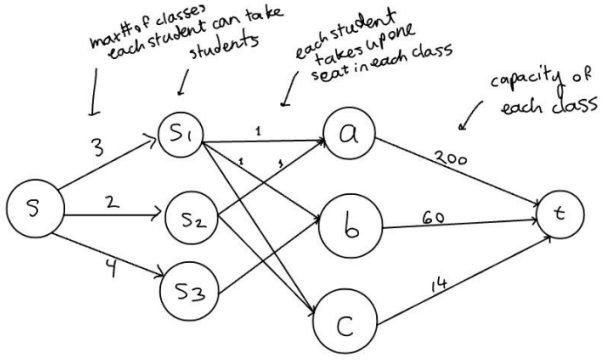
   (a) Consider the flow network G' obtained by adding 1 to the capacity of e. We will prove that the value of the maximum flow in G' is either (f) or $(f) + 1$. Proof: By the Max-Flow Min-Cut theorem, we know there will exist some cut in (A,B) in G that has a capacity of $v(f)$. What is the capacity of the cut (A,B) in G'? If edge e is in the cut (A,B), the capacity of (A,B) will increase by 1, so G' can now have a flow of at most $v(f) + 1$.

   (b) Show how to find a maximum flow in the resulting capacitated graph in time $O(m + n)$, where m is the number of edges in G and n is the number of nodes. First, create a residual graph G'_f, which is a graph that contains the remaining capacities of each edge of G', after removing the flow assignments

Example

Student	m	list of classes
S 1	3	{a, b, c}
S 2	2	{a, c}
S 3	4	{b}

class	capacity
a	200
b	60
c	14



from G. Think of it as G' - the flow assignment of $G = G'_f$. Edges that have no more remaining capacity $c_e = 0$ will not be present in G'_f. Use BFS or DFS to see if there is a path from $s - t$. If there is, then the maximum flow can be increased by 1.

Recommended Video:

- Ford-Fulkerson in 5 minutes, Micheal Sambol, `https://youtu.be/Tl90tNtKvxs?si=1FPRBdobpX1d1eEM`