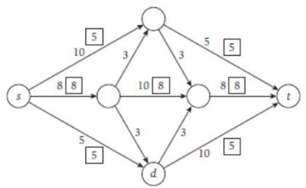
CS 577 Assignment 9 – Network Flow Fall 2022

Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.

Name: Ramapriya R Wisc id: rranganath

Network Flow

1. *Kleinberg, Jon. Algorithm Design (p. 415, q. 3a)* The figure below shows a flow network on which an $s-t$ flow has been computed. The capacity of each edge appears as a label next to the edge, and the flow is shown in boxes next to each edge. An edge with no box has no flow being sent down it.

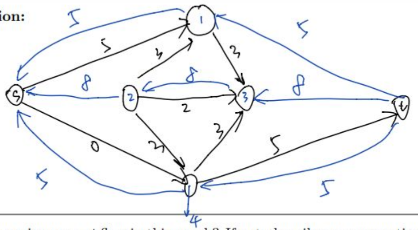


(a) What is the value of this flow?

Solution: $5+8+5 = 18$

(b) Please draw the **residual graph** associated with this flow.

Solution:



(c) Is this a maximum $s-t$ flow in this graph? If not, describe an augmenting path that would increase the total flow.

Solution: s → 1 → 3 → 2 → 4 → t flow is 3

2. *Kleinberg, Jon. Algorithm Design (p. 418, q. 8)* Consider this problem faced by a hospital that is trying to evaluate whether its blood supply is sufficient:

   In a (simplified) model, the patients each have blood of one of four types: A, B, AB, or O. Blood type A has the A antigen, type B has the B antigen, AB has both, and O has neither. Patients with blood type A can receive either A or O blood. Likewise patients with type B can receive either B or O type blood. Patients with type O can only receive type O blood, and patients with type AB can receive any of the four types.

   (a) Let s_O, s_A, s_B, s_{AB} denote the hospital's blood supply on hand, and let d_A, d_B, d_O, d_{AB} denote their projected demand for the coming week. Give a polynomial time algorithm to evaluate whether the blood supply is enough to cover the projected need.

Solution:

Input. s_O. s_A. s_B. s_{AB}. d_O. d_A. d_B. d_{AB}

Output: boolean variable.

if $s_O < d_O$ return false

if $s_A + s_O - d_O < d_A$ return false

if $s_B + s_O - d_O < d_B$ return false

if $s_A + s_B + s_O - d_O < d_A + d_B$ return false.

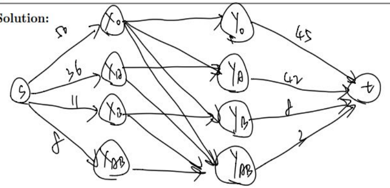
if $s_{AB} + (s_A + s_B + s_O - d_O - d_A - d_B) < d_{AB}$ return false.

return true.

   (b) Network flow is one of the most powerful and versatile tools in the algorithms toolbox, but it can be difficult to explain to people who don't know algorithms. Consider the following instance. Show that the supply is **insufficient** in this case, and provide an explanation for this fact that would be understandable to a non-computer scientist. (For example: to a hospital administrator.) Your explanation should not involve the words *flow*, *cut*, or *graph*.

blood type	supply	demand
O	50	45
A	36	42
B	11	8
AB	8	3

Solution:



Intuitively, our supply can not fullfill demand A. A can only receive O, B. demand A must gain $42 - 36 = 6$ from supply O. but supply O can only give $50 - 45 = 5 < 6$ after fullfolling itself demand.

3. *Kleinberg, Jon. Algorithm Design (p. 419, q. 10)* Suppose you are given a directed graph $G = (V, E)$. This graph has a positive integer capacity c_e on each edge, a source $s \in V$, a sink $t \in V$. You are also given a maximum $s - t$ flow through G: f. You know that this flow is *acyclic* (no cycles with positive flow all the way around the cycle), and every flow $f_e \in f$ has an integer value.

   Now suppose we pick an edge e^* and reduce its capacity by 1 unit. Show how to find a maximum flow in the resulting graph G^* in time $O(m + n)$, where $n = |V|$ and $m = |E|$.

Solution: The maximum flow $\to M$ can stay unchanged or decrease by 1. Since we have f

① If $f(e^*) < C_{e^*}$ then stay unchange.

② If $f(e^*) = C_{e^*}$, find a flow one path from t to s go through e^* in residual graph of f.

. More specifically, consider $e^* = (u, v)$. let $f'(e^*) = f(e^*) - 1$. then find flow one path from $u \to s$, $t \to v$ $O(M+n)$

adjust f to f'. Now we have flow f' on new graph with flow $M-1$

Now we try to find some augmenting path from $s - t$ with new residual graph of f' $O(m+n)$

If we can find, max flow is still M. if not, we have $M-1$.

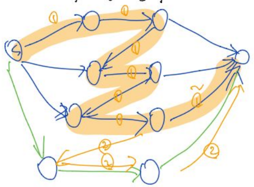
4. *Kleinberg, Jon. Algorithm Design (p. 420, q. 11)* A friend of yours has written a very fast piece of code to calculate the maximum flow based on repeatedly finding augmenting paths. However, you realize that it's not always finding the maximum flow. Your friend never wrote the part of the algorithm that uses backward edges! So their program finds only augmenting paths that include all forward edges, and halts when no more such augmenting paths remain. (Note: We haven't specified *how* the algorithm selects forward-only augmenting paths.)

   When confronted, your friend claims that their algorithm may not produce the maximum flow every time, but it is guaranteed to produce flow which is within a factor of b of maximum. That is, there is some constant b such that no matter what input you come up with, their algorithm will produce flow at least $1/b$ times the maximum possible on that input.

   Is your friend right? Provide a proof supporting your choice.

Solution: My friend is false in general

Consider we following graph with all edges 1



My friend would find

① → ① → ·· ① to have

solution 1.

But in blue graph best

would be 3.

we can add infinite new

one (green path) to have

higher maximum flow while

my friends to find

① → ① ··· ② → ② → ②.

still 1.

So we can not find constant b

to let it produce at least $\frac{1}{b}$ of