COMP 360 - Fall 2019 - Assignment 1

Due: 11:59pm Sept 24th.

General rules: In solving these questions you may consult books or other available notes, but you need to provide citations in that case. You can discuss high level ideas with each other, but each student must find and write his/her own solution. Copying solutions from any source, completely or partially, allowing others to copy your work, will not be tolerated, and will be reported to the disciplinary office.
You should upload the pdf file (either typed, or a clear and readable scan) of your solution to mycourses.
There are in total 27 points, but your grade will be considered out of 20, however no partial marks are given for the last bonus question

"Oh no, not homework again."

---

1. There are 169 students who are in need of emergency treatment. Each of these students requires a transfusion of one unit of whole blood. The clinic has supplies of 170 units of whole blood. The number of units of blood available in each of the four major blood groups and the distribution of patients among the groups is summarized below.

Blood type	A	B	O	AB
Supply	46	34	45	45
Demand	39	38	42	50

Type A patients can only receive type A or O; type B patients can receive only type B or O; type O patients can receive only type O; and type AB patients can receive any of the four types.

   (a) (1 points) Give a max flow formulation that determines a distribution that satisfies the demands of a maximum number of patients (Explain why the solution to your formulation corresponds to the problem). Draw the directed graph and put the edge capacity above each edge. Your network should have 10 vertices: a source (named s), a supply node for each of the four blood types (named a, b, o, ab), a demand node for each blood type (named a', b', o', ab'), and a sink (named t).

   (b) (2 points) Solve the maximum flow problem using the Ford-Fulkerson algorithm. Do the first augmentation on the path $s - b - ab' - t$. List each of the augmenting paths below, and state the amount by which the path increases the current value of the flow. Also, write the final flow values on each edge in the network above.

(c) (1 points) Find a minimum cut in the network above, i.e. list the vertices on the source side of the cut.

2. (3 points) Consider a flow network on 10 vertices (including the source and the sink), consisting of a single path starting at a source s and ending at a sink t. The capacities of all edges are equal to 1. How many cuts does this flow network have? How many of them are min-cuts?

3. (3 points) In the class, we discussed why the naive Ford-Fulkerson algorithm is not an efficient algorithm. In this question, we want to expand on the details. Recall that a polynomial-time (a.k.a. efficient) algorithm is one that runs in $O(N^c)$ where c is a fixed constant[1], and N is the number of bits required to represent the input.

   Suppose that the input to the Ford-Fulkerson algorithm is given in the following format in a file. First the number of vertices $n \geq 2$ is written in the file, followed by a space, and then followed by $n \times n = n^2$ numbers that are separated by spaces. Here these n^2 numbers represent an $n \times n$ matrix A whose ij-th entry A_{ij} equals to the capacity of the edge from i to j (it is 0 if such an edge is not present in the flow network). Suppose all these numbers are represented in decimal using the 7-bit ASCII characters (for each decimal digit). We assume that the 1-st vertex is the source and the n-th vertex is the sink.

   Pick $N \geq 1000$ that is divisible by 100, and let the input be the network with vertices s, a, b, t, and edges sa, sb, at, bt with capacities $10^{N/100}$ and the edge ab with capacity 1.

   - Show that the input size[2], i.e. the input file is smaller than N bits.
   - Show that the running time of the Ford-Fulkerson on this input can be longer than $2^{N/100}$, and thus conclude that the naive FF is of exponential time and thus is not an efficient algorithm.

4. (2 Points) Show that for every even m, there is a flow networks that requires $m/2$ augmentations no matter how we choose the augmenting paths. Here m is the number of the edges of G. (If you are not familiar with the notation, see the definition of big Omega in Chapter 2.2 of the textbook).

5. (3 points) The goal of this exercise is to show that if we modify the Ford-Fulkerson algorithm so that it does not decrease the flow on any of the edges (i.e. we do not add the backward edges to the residual graph), then the algorithm might perform very poorly.

   Show that there exists a flow network whose maximum flow is equal to 1000, but the above modified Ford-Fulkerson algorithm terminates after finding a flow of value 1. In other words, in your flow network, there is an augmenting path with bottleneck 1 such that after augmenting this path, it is not possible to augment the flow without cancelling some flow on one of the edges.

6. (2 points) Consider the modified Ford-Fulkerson algorithm from the previous question (i.e. FF without backward edges in the residual graph), but now suppose that we always pick the shortest such augmenting path. Does this algorithm find the max flow? If yes, present a proof, otherwise give an example where it fails to produce the max flow.

7. (2 points) Recall that for every flow f and every cut (A, B), we have $\mathrm{val}(f) = f^{out}(A) - f^{in}(A)$ where $f^{out}(A) = \sum_{\substack{uv \in E \\ u \in A, v \in B}} f(uv)$ and $f^{in}(A) = \sum_{\substack{uv \in E \\ u \in B, v \in A}} f(uv)$. Use this fact to prove that $\mathrm{val}(f)$ is equal to the sum of the flow on the edges going into the sink.

[1] for example $O(N^3)$, or $O(N^{100})$, but not $O(N^{\log(N)})$

[2] Recall that each ASCII character is 7 bits

8. (3 points) Suppose that we have solved the Max Flow problem on a flow network $(G = (V, E), s, t, \{c_e\}_{e \in E})$, and found the flow $f : E \to \mathbb{R}$ with the largest value. Someone gives us the opportunity to increases the capacity of one of the edges by 1. We want to choose this edge carefully so that to increase the value of the max-flow. How can we find such an edge in $O(m)$? Give an example where such an edge does not exist.

9. (Bonus question: No partial marks are given for this question) Suppose that we allow irrational capacities. Consider the flow network with nodes s, t, a, b, c, d, and edges sa, sc, sd with capacity 4, and at, bt, dt with capacity 4, and cd and cb with capacity 1, and ab with capacity $\frac{\sqrt{5}-1}{2}$ (which is a solution to $1 - x = x^2$).

   - (0 points) What is the maximum flow on this network?
   - (5 points) Let P_0 be the augmenting path $scbt$, P_1 be the augmenting path $sabcdt$, P_2 be the augmenting path $scbat$, and P_3 be the augmenting path $sdcbt$. Show that if FF uses the following sequence of paths

     $$P_0, (P_1, P_2, P_1, P_3), (P_1, P_2, P_1, P_3), (P_1, P_2, P_1, P_3) \ldots$$

     it will never terminate, and in fact never finds any flow with value larger than 7.