CS180 Homework 7

1. In class, we have roughly discussed how to implement the idea by Siddharth to solve the midterm water flow problem. His idea is creating the pressure function committing sections growing of it by considering intervals in descending order of flow. Whatever we committed will never be changed.

   We first create an array of the 2n breakpoints which are either the start of an interval or the end of an interval. We think of each entry of this array as a pair: A value of a break point, followed by a pointer to the next break point "of interest."

   Initially a pointer of entry j in the array points to break point in position $j + 1$ in the array. And the graph now at the beginning is of height 0 between break points.

   The observation of Siddharth is that now we take the interval with the max flow demand say *maxflow1*. Say it starts at break point i and ends at break point $j > i$. Then the observation is that the final graph will have value *maxflow1* between break points i and j.

   So now, we would like to have the pointer at i point to j, and now continuing inductively. This will be easy if intervals are disjoint. But they are not. So the next interval in the order of flow with *maxflow2* may start say between i and j but end after at $k > j$. So now we would like to have the pointer in i point now to k because now we committed the graph between i and k.

   But how do we know that the beginning of the interval of *maxflow2* is in between i and j unless we "memorize" i and j which will be expensive!

   So, we want a mechanism that if a new interval has a start or a finish falling in a committed segment we can not too expensively find this segment.

   Here comes handy the Union-Find algorithm discussed in earlier homework (review). At the beginning, each break point is a tree consisting of itself. When we put in the first interval of *maxflow1* we Union all the break points between i and j, to a rooted tree that includes all of them. Now when i take the beginning of interval *maxflow2* I just follow that breakpoint to its root. Its root tells me the interval it belongs to. The idea is that when we Union we hang small height tree off the big one, so that going from any node in a tree to its root is less that $\log 2n$ steps.

   Develop this idea into an algorithm and argue its complexity is $O(n \log n)$. And if Siddharth worked the details of his idea correctly on the midterm...Prof Gafni will do an "end-zone NFL dance" for him.

2. In an undirected graph $G = (V, E)$, the edge s-t connectivity is the minimum number of edges to disconnect two vertices s and t. Using max flow, prove that the s-t connectivity equals the maximum number of edge disjoint paths between s and t. (Hint: replace G by creating parallel edges and assign edge capacities 1. This observation is called Menger Theorem and will help you copying from the book or the internet if you are so inclined.)

3. The s-t vertex-connectivity of undirected graph $G = (V, E)$, is the minimum number of nodes in order to disconnect two vertices s and t. Show that the s-t vertex-connectivity equals the number of vertex-disjoint paths between s and t. (Hint: split a vertex into two vertices)

4. Suppose we are given the maximum flow in a flow network $G = (V, E)$ with source s, sink t, and integer capacities. Suppose the capacity of a single edge is increased by one. Give an $O(n + m)$ algorithm for updating the maximum flow, where G has n vertices and m edges.

★ Express your algorithm in a well-structured manner. Use pseudo code and the textbook has good examples to follow. Avoid using a long continuous piece of text to describe your algorithm. Start each problem on a NEW page. Unless specified, you should justify the time complexity of your algorithm and why it works. For grading, we will take into account both the correctness and the clarity. Your answers are supposed to be in a simple and understandable manner and sloppy answers are expected to receiver fewer points.

★ Homework assignments are due on Gradescope. Email attachments or paper submissions are NOT acceptable.

★ Raw photo is NOT acceptable. Upload your homework as a PDF scan by using a scanner or mobile scanner app. Match each problem with your answer on Gradescope. Use dark pen or pencil and your handwriting should be clear and legible.

★ We recommend using LaTeX, LyX or other word processing software for writing the homework. This is **NOT** a requirement but it helps us to grade and give feedback.