ASSIGNMENT 1
CSC 425, FALL 2018

Due Thursday Oct 19 at 11:55 PM online. Based on readings on Fully dynamic algorithms and distributed computing

Homework policy: See homework 1.

(1) Find Dinic's Algorithm on Wikipedia, Example 1. Suppose the first blocking flow computation is implemented with link-cut trees:
(a) Draw a picture of the forest of link-cut trees after the first flow is sent, and after the second flow is sent.
(b) What link-cut operations are performed to send the first flow? The second flow? (Name the edge or path they are associated with.)

(2) **Software company problem.** We are starting a company and are deciding which software packages to offer. Each software package requires certain programs, and programs may be used for more than one software package. Let $W = \{W_1, W_2, ..., W_m\}$ be the set of possible software packages.. Each software package will bring in p_j revenue. Each software package W_i uses a subset R_i of the set $I = \{I_1, I_2, ..., I_n\}$ of programs. The cost of coding the program I_k is c_k dollars. Design an efficient algorithm to determine which programs to code and products to produce, in order to maximize the net revenue, which is the total revenue from software packages minus the total cost of all programs coded.

Consider the following network G. The network contains a source vertex s, vertices $I_1, I_2, ..., I_n$, vertices $W_1, W_2, ..., W_m$, and a sink vertex t.

For $k = 1, 2..., n$, there is an edge (s, I_k) of capacity c_k, and for $j = 1, 2, ..., m$, there is an edge (W_j, t) of capacity p_j.
For $k = 1, 2, ..., n$ and $j = 1, 2, ..., m$, if $I_k \in R_j$, then there is an edge (I_k, W_j) of infinite capacity.

a. Show that if $W_j \in T$ for a finite-capacity cut (S, T) of G, then $I_k \in T$ for each $I_k \in R_j$.

b. Show how to determine the maximum net revenue from the capacity of the minimum cut of G and the given p_j values.

A full binary tree is a tree in which every node other than the leaves has two children.

c. Give an efficient algorithm to determine which packages to produce and which programs to code. Analyze the running time of your algorithm in terms of m, n ,and $r = \sum_{j=1}^{m} |R_j|$.

mst doable

(3) Given $G = (V, E)$, suppose E is partitioned into E_1, E_2. Let $MST(G)$ denote the minimum spanning tree (or forest if not connected) of G.
a) Prove that the $MST(G) = MST(MST(V, E_1) \cup MST(V, E_2))$. Hint: Use the blue rule.
b) Suppose that we build a full binary tree as follows: We arbitrarily partition the edges of E into n sets of size $\leq n$ and assign the set to a leaf and compute the MST of each leaf. Each parent compute the MST's of the union of the MST of its two children, and this is done up to the root. Explain why the root contains the MST of the whole graph.
c) Give an algorithm which uses this data structure to do edge inserts and deletes in a dynamic MST in worst case time per insertion or deletion $O(nlog^2n)$. Hint: you may assume that computing MST from scratch takes $O(m \log n)$ time with n nodes and m edges. Explain why this is your running time.

Minimum spanning forest is a generalization of minimum spanning tree for unconnected graphs.

For every component of the graph, take its MST and the resulting collection is a minimum spanning forest.

A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far as possible.

(4) a) How are the edges of the BFS (breadthfirst search) tree picked using Bellman-Food?
b) How can Bellman-Ford be modified so that it terminates after the BFS is constructed (in an asynchronous network), without increasing its cost and time by more than a constant factor? Explain.

(5) Suppose the nodes in an asynchronous network know the IDs of their neighbours and suppose that messages can hold up to n IDs. Give an algorithm which finds the BFS tree in $O(D^2)$ time and $O(nD)$ messages.