CS 561, HW 12

Prof. Jared Saia, University of New Mexico

1. Exercise 23.2-1 in CLRS, "Run the Floyd Warshall algorithm ..."

2. **Recursion Cat** You are given a tree with all nodes colored either red or black. Call a path *valid* if at any step of the path, the number of red nodes visited so far is greater than or equal to the number of black nodes visited so far. A cat starts at the root node of the tree and wants to find a valid path to some leaf node.

   For each node v, let $f(v)$ be $-\infty$ if there is no valid path to v. Otherwise, let $f(v)$ be the number of red nodes visited minus the number of black nodes for the path ending at v.

   (a) Give a recurrence relation for f. Hint: you may find it useful to let $p(v)$ be the parent of v, for every node v that is not the root.

   (b) Briefly describe a dynamic program that uses the recurrence above to return a valid path from root to some leaf, if such a path exists.

   Now recursion cat wants to find valid paths on any graph. Define a *red cycle* to be a cycle that has more red than black nodes in it. Assume you are given a graph, G, with no red cycles. For any pair of nodes, you want to determine if there is a valid path from u to v. Taking inspiration from Floyd-Warshall, you first assign labels 1 to n to all n nodes in the graph. Then you consider paths from nodes u to v that visit intermediate nodes with label at most i. For a given path, let the *black excess* of that path be the maximum over all steps of the path of the number of black nodes minus the number of red nodes at any step. For example, a path of the form R, B, R, B, B, B, R, R has black excess of 2.

   Define $f(u, v, i, b) = -\infty$ if there is no path from u to v using intermediate nodes of label at most i, with black excess at most b. Otherwise, define $f(u, v, i, b)$ to be the maximum, over all paths from u to v, with black excess at most b that visit intermediate nodes with label at most i, of the number of red nodes minus the number of black nodes in that path. For example, if the only path from u to v has form R, B, R, B, B, B, R, R, then $f(u, v, n, 2) = 0$.

(c) Write a recurrence relation for $f(u, v, i, b)$. It may help to assume that $-\infty + x = -\infty$ for any value x. Hint: Let the base case(s) be $f(u, v, 0, b)$ for any values of u, v and any b, $0 \le b \le n$. It may help to define for a node v, $color(v)$ to be 1 if the node is red, and -1 if the node is black.

(d) Briefly describe a dynamic program that uses the recurrence above to determine if a valid path exists from u to v for every u and v. What is the runtime as a function of n, the number of nodes, and m the number of edges?

3. The **Subgraph Isomorphism** problem takes as input two undirected graphs G_1 and G_2 and returns TRUE iff G_1 is isomorphic to a subgraph of G_2. Prove that the Subgraph Isomorphism problem is NP-Complete.

4. In the MIN-INSIDE-EDGES problem, you are given a graph $G = (V, E)$, and a number $x \le |V|$, and you must choose a subset $V' \subseteq V$ of size x. Call an edge in E *inside* if both endpoints of the edge are nodes in V'. Your goal is to output the minimum number of inside edges for any set V' of size x.

   (a) Show that MIN-INSIDE-EDGES is NP-Hard by a reduction from one of the following: 3-SAT, VERTEX-COVER, CLIQUE, SUBGRAPH-ISOMORPHISM, INDEPENDENT-SET, 3-COLORABLE, HAMILTONIAN-CYCLE, or TSP.

   (b) Consider the randomized algorithm that picks a subset V' of size x, uniformly at random from all subsets of V of size x, when given graph $G = (V, E)$ and number x. Compute the expected number of inside edges for this algorithm using indicator random variables and linearity of expectation. Let $n = |V|$ and $m = |E|$.

(c) Let μ be the expected number of inside edges for the randomized algorithm (i.e. your answer from part (b)). Now use Markov's inequality to bound the probability that there are greater than or equal to $(11/10)\mu$ inside edges after running the algorithm in part (b).

(d) Using your result from part (c), bound the expected number of times you would need to run the randomized algorithm before you get a solution that has less than $(11/10)\mu$ inside edges. *Hint: Recall that for a random variable Y taking on positive integer values, $E(Y) = \sum_{i=1}^{\infty} Pr(Y \geq i)$ (see slides 32-33 from our "Randomized Data Structures" lecture).*