CSE421: Design and Analysis of Algorithms April 14, 2021

Homework 3

Shayan Oveis Gharan Due: April 22, 2021 at 23:59 PM

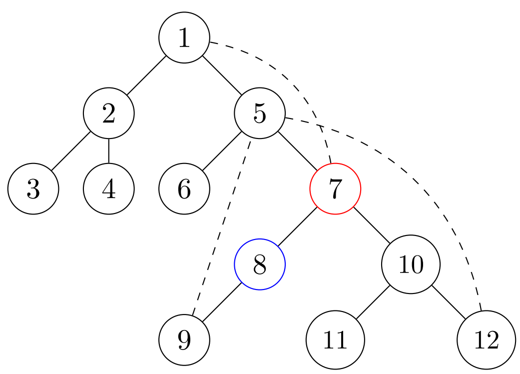
P1) (20 points) The main goal of this question is to answer the problem in part (d). If you prefer you can ignore parts (a,b,c) and solve (d) directly.

a) **Optional 0 points** Let T be a tree with n vertices. Prove that there is a **unique** path in T between each pair of vertices.

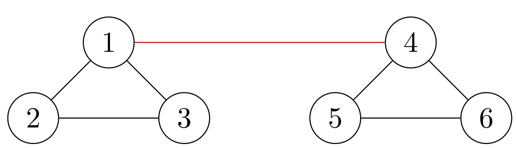
b) **Optional 0 points** Let $G = (V, E)$ be a connected undirected graph and $T \subseteq E$ be a spanning tree of G. Prove that for any edge $e \in T$, $G - e$ is connected iff there is an edge $f = (u, v) \in E - T$ such that the unique path between u, v in T has the edge e.

c) (10 points) Show how to modify the code for recursive depth-first search of undirected graphs to obtain an $O(n + m)$ time algorithm that (i) assigns each node v a number, dfsnum(v), indicating a sequence number for when v was first visited by DFS, and computes $min(v)$ for each node v, the smallest dfsnum of any node that was encountered in the recursive call $dfs(v)$.

For example, in the following picture edges of the DFS tree are marked in solid (and non-tree edges in dashed). Every node is labelled with its dfsnum. So, $min(.)$ for the red node is 1 and min of the blue node is 5.



d) (10 points) Given a graph $G = (V, E)$ with n vertices and m edges, design an $O(m + n)$ time algorithm that for any edge $e \in E$ outputs if $G - e$ is connected. For example, given the following graph you should output "yes" for all black edges and "no" for the red edge.



P2) (20 points) Prove or disprove: Every directed graph with n vertices and at least $n(n-1)/2+1$ directed edges has a cycle.

P3) Amy, a TA in 421 is asked to grade n sheets of exams, $1, \ldots, n$. Let t_i be the time that takes her to grade the i-th sheet (perhaps, t_i depends on how long the i-th proof is). Whenever she grades the i-th sheet the grade will be published in Gradescope right away. Say she finishes grading i-th sheet at time f_i. Then, the average grading time of all exams is $\frac{1}{n}(f_1 + \cdots + f_n)$. To make the students happy, Shayan asked Amy to minimize the average grading time, that is the average time for a 421 student to see his/her grade. We want to help Amy by figuring out the optimal order to grade these sheets. Design an efficient algorithm which outputs the minimum average grading time (note that your algorithm just needs to output a number).

For example, if $t_1 = 3, t_2 = 2, t_3 = 4$ then the optimal order to grade is, $2, 1, 3$. Then, we have $f_2 = 2, f_1 = 2 + 3 = 5, f_3 = 2 + 3 + 4 = 9$ and the average grading time is $16/3$.

P4) You get interested in investing in the stock market. You are so lucky to have a friend who knows the price of a specific commodity in the next n days; say the price will be p_i in day i and it is given to you in the input. Every day you can either buy the commodity (if you don't have it already) or sell it (if you have it). Note that there is a single commodity in this market and you cannot buy two copies of it. Design an $O(n)$ time algorithm which outputs the maximum possible gain.

For example, consider the following prices in the next $n = 10$ days.

Day	1	2	3	4	5	6	7	8	9	10
Price	11	7	10	9	13	14	10	15	12	10

The optimum startegy is to do as follows:

- Buy on day 2 and sell on day 3 (gain +3)
- Buy on day 4 and sell on day 6 (gain +5)
- Buy on day 7 and sell on day 8 (gain +5)

So, your total gain is 13 and you should output 13.

P5) **Extra Credit:** Suppose G is a 3-colorable graph with n vertices, i.e., it is possible to color the vertices of G with 3 colors such that the endpoints of every edge have distinct colors. Design a polynomial time algorithm that colors vertices of G with $O(\sqrt{n})$ many colors.