CS 6363.500 Homework 3

Due Thursday March 14th on eLearning

February 28, 2019

Please answer each of the following questions.

1. Recall the optimal binary search tree problem. The input consists of a sorted array $A[1 .. n]$ of search keys and an array $f[1 .. n]$ of frequency counts, where $f[i]$ is the number of searches for $A[i]$. The cost of a given binary search tree T for $A[1 .. n]$ is

$$Cost(T, f[1 .. n]) = \sum_{i=1}^{n} f[i] \cdot \#\text{ancestors of } v_i \text{ in } T$$

   where v_i is the node that stores $A[i]$.

   AVL trees were the earliest self-balancing binary search trees, first described in 1962 by Georgy Adelson-Velsky and Evgenii Landis. An AVL tree is a binary search tree where for every node v, the height of the left subtree of v and the height of the right subtree of v differ by at most one.

   Describe and analyze a dynamic programming algorithm to construct an optimal AVL tree for a given set of search keys and frequencies. An algorithm that computes the cost of the optimal AVL tree is worth full credit.

2. You've just been appointed as the new organizer of Giggle, Inc.'s annual mandatory holiday party. The employees at Giggle are organized into a strict hierarchy, that is, a tree with the company president at the root. Each employee is to be given a present at the party, and your job is to decide which present each employee receives.

   Specifically, each employee must receive one of three gifts: (1) an all-expenses-paid six-week vacation anywhere in the world, (2) an all-the-pancakes-you-can-eat breakfast for two at Jumping Jack Flash's Flapjack Stack Shack, or (3) a used-up dry erase marker. Corporate regulations prohibit any employee from receiving exactly the same gift as his/her direct supervisor. Any employee who receives a better gift than his/her direct supervisor will almost certainly be fired in a fit of jealousy.

   Describe and analyze a dynamic programming algorithm to distribute gifts so that the minimum number of people get fired. Yes, you may send the president a used-up marker.

   More formally, you are given a rooted tree T, representing the company hierarchy, and you want to label the nodes of T with integers 1, 2, or 3, so that every node has a different label from its parents. The *cost* of a labeling is the number of nodes with smaller labels than their parents. See Figure 1 for an example. You need to describe and analyze an algorithm to compute the minimum-cost labeling of T. Again, computing the minimum cost itself is sufficient to answer the question.

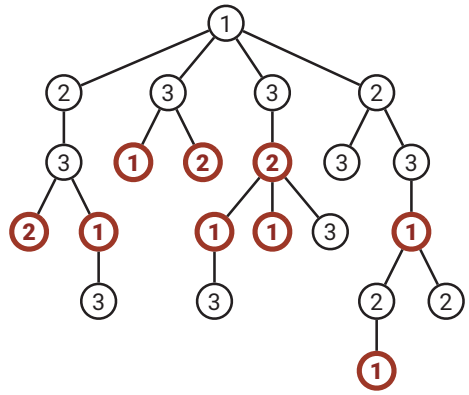

Figure 1. A tree labeling with cost 9. The nine bold nodes have smaller labels than their parents. This is *not* the optimal labeling for this tree.

3. Consider the following process. At all times you have a single positive integer x, which is initially equal to 1. In each step, you can either *increment* x, or *double* x. Your goal is to produce a target value n. For example, you can produce the integer 10 in four steps as follows:

$$1 \xrightarrow{+1} 2 \xrightarrow{\times 2} 4 \xrightarrow{+1} 5 \xrightarrow{\times 2} 10$$

Obviously you can produce any integer n using exactly $n-1$ increments, but for almost all values of n, this is horribly inefficient. Describe and analyze an algorithm to compute the *minimum* number of steps required to produce any given integer n. If you use a greedy algorithm, you must give a formal proof of correctness including the necessary exchange argument.

4. Consider a directed graph G, where each edge is colored either red, white, or blue. A walk in G is called a *French flag walk* if its sequence of edge colors is red, white, blue, red, white, blue, and so on. More formally, a walk $v_0 \to v_1 \to \ldots \to v_k$ is a French flag walk if, for every integer i, the edge $v_i \to v_{i+1}$ is red if $i \mod 3 = 0$, white if $i \mod 3 = 1$, and blue if $i \mod 3 = 2$. French flag walks *are* allowed to repeated vertices.

Describe an algorithm to find all vertices in G that can be reached from a given vertex v through a French flag walk. *[Hint: Perform a* reduction *to normal graph reachability.]*

5. Let G be a directed acyclic graph whose vertices have labels over some fixed alphabet, and let $A[1 .. \ell]$ be a string over the same alphabet. Any directed path in G has a label, which is a string obtained by concatenating the labels of its vertices.

   (a) Describe and analyze a dynamic programming algorithm that correctly determines if there is a path in G whose label is A.

   (b) Describe and analyze a dynamic programming algorithm to find the *number* of paths in G whose label is A. (Assume that you can add arbitrarily large integers in $O(1)$ time.)