CS140 - Group Assignment 7

Due: Friday, Oct. 21 at 11:59pm

Use this group session to help get ready for the checkpoint next week. I've included some practice problems below or you can also review some concepts that you're a little fuzzy on.

1. Practice Questions

   Below are some practice problems to help give you study for the upcoming checkpoint. Note that not all of these would necessarily be good exam problems, but are there to provide you with some additional practice on the materials.

   (a) Describe a red-black tree on n values that has the largest possible ratio of red internal nodes to black internal nodes. What is this ratio? What tree has the smallest possible ratio, and what is the ratio?

   (b) Consider the following arrays. Draw each as an almost-complete binary tree (i.e. a heap). Is the tree a max-heap, a min-heap, or neither?

      i. $[4, 12, 8, 21, 14, 9, 17]$

      ii. $[701, 253, 24, 229, 17, 22]$

   (c) Suppose we perform a sequence of n operations on a data structure in which the ith operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

   (d) A student wants to walk up a staircase with n steps. They can skip up to 2 steps at a time: in other words, they can go from step k to step $k+1$, they can skip one step and go from step k directly to step $k+2$, or they can skip 2 steps and go from step k directly to step $k+3$. Let $Num(i)$ be the number of ways that the student can get to step i. Write and justify a recursive formula for $Num(i)$. If you were to implement your solution as an iterative dynamic program, what would the pseudocode be? What are bounds on the space and time requirements of your dynamic programming algorithm?

   (e) Change revisited

      In class we discussed the change problem and, in particular, proved that a greedy strategy was optimal for US denominations (penny=1, nickel=5, dime=10, quarter=25).

      The change problem in general can be specified as: make change for an amount of money C with as few coins as possible for coin denominations with values $v_1 > v_2 > \ldots > v_n$ (all integers), where $v_n = 1$.

      i. The greedy approach only works for certain coin values. Given an example of coin denominations and a target amount such that the greedy strategy does not provide the optimal solution.

ii. Give a dynamic programming solution for the function which calculates the minimum number of coins necessary to make the amount. Make sure you give both a written description of what's stored in the dynamic programming table as well as the recursive definition.

iii. What is the size of your dynamic programming table? What entry contains the answer? What is the running time of your algorithm?

iv. Fill out the dynamic programming table assuming the denominations are $v_1 = 6, v_2 = 5, v_3 = 1$ and the value $C = 10$.

(f) T/F: There is no way to insert n elements into a hashtable of size m where $n > m$.

(g) T/F: If two values x, y are different (i.e., $x \neq y$) then there is no way for them to have the same hash function value, i.e., $h(x) = h(y)$.

(h) T/F: In a binary search tree, ignoring the root, it is not possible for duplicate values to be in different halves of the tree, i.e., one value in root.left and one in root.right.

(i) T/F: In a binary search tree, the root cannot have a successor.

2. Group participation: Who attended the group meeting?