CS210a Data Structures and Algorithms
Assignment 3
Due date: October 24 at 11:59 pm
Total of 20 Marks

这次不用写在纸上

Submit through OWL a pdf file or an image file with your answers to the following questions. No hard copy of your answers is required for this assignment. Remember that no late concept assignments will be accepted.

1. (2 marks) Consider a hash table of size $M = 7$ where we are going to store integer key values. The hash function is $h(k) = k \bmod 7$. Draw the table that results after inserting, **in the given order**, the following key values: 18, 11, 12, 47, 22. Assume that collisions are handled by separate chaining.

2. (2 marks) Show the result of the previous exercise, assuming collisions are handled by linear probing.

3. (2 marks) Repeat exercise (1) assuming collisions are handled by double hashing, using secondary hash function $h'(k) = 5 - (k \bmod 5)$.

4. (3.5 marks) Consider the following algorithm.

   Algorithm $foo(n)$
   if n = 0 **then return** 1
   else {
   $x \leftarrow 0$
   for $i \leftarrow 1$ **to** n **do** $x \leftarrow x + x/i$
   $x \leftarrow x + foo(n-1)$
   return x
   }

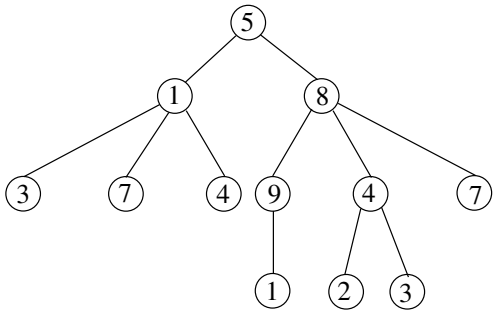
   The time complexity of this algorithm is given by the following recurrence equation:

   $f(0) = c_1$
   $f(n) = f(n-1) + c_2 n + c_3$, for $n > 0$

   where c_1, c_2, and c_3 are constants. Solve the recurrence equation **and** give the value of $f(n)$ and its order using big-Oh notation. You **must** explain how you solved the recurrence equation.

5.(i) (7 marks) Write in pseudocode an algorithm `maxValue(r)` that receives as input the root r of a tree (not necessarily binary) in which every node stores an integer value and it outputs the largest value stored in the nodes of the tree. For example, for the following tree the algorithm must output the value 9.

写一个找node最大数值的程序



For a node v use v.`value` to denote the value stored in v; v.`isLeaf` has value true if node v is a leaf and it has value false otherwise. To access the children of a node v use the following pseudocode:

for each child c of v **do**

5.(ii) (3.5 marks) Compute the worst case time complexity of your algorithm as a function of the total number n of nodes in the tree. You must

- explain how you computed the time complexity
- give the order of the time complexity of the algorithm