4190.407(001): ALGORITHM
Homework #2

Due on October 29, 2020

Professor Sun Kim

Minghang Li (2017-17232)

October 29, 2020

Problem 1

Consider inserting the keys $10, 22, 31, 4, 15, 28, 17, 88, 59$ into a hash table of length $m = 11$ using open addressing with the auxiliary hash function $h'(k) = k$. Illustrate the result of inserting these kes using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \mod (m-1))$.

Solution

Problem 2

An alternative method of performing an inorder tree walk of an n-node binary search tree finds the minimum element in the tree by calling $TREE-MINIMUM$ and then making $n-1$ calls to $TREE-SUCCESSOR$. Prove that this algorithm runs in $\Theta(n)$ time.

Solution

Problem 3

Write a nonrecursive version of $OS - SELECT$.

Solution

Problem 4

Draw the recursion tree for the $MERGE-SORT$ procedure form Section 2.3.1 on an array of 16 elements. Explain why memoization fails to speed up a good divide- and-conquer algorithm such as $MERGE-SORT$.

Solution

Problem 5

Determine an LCS of $< 1, 0, 0, 1, 0, 1, 0, 1 >$ and $< 0, 1, 0, 1, 1, 0, 1, 1, 0 >$.

Solution

Problem 6

A **palindrome** is a nonempty string over some alphabet that reads the same forward and backward. Examples of palindromes are all strings of length 1, *civic*, *racecar* and *aibohphobia* (fear of palindromes).
Give an efficient algorithm to find the longest palindrome that is a subsequence of a given input string. For example, given the input *character*, your algorithm should return *carac*. What is the running time of your algorithm?

Solution