Algorithms -1
Tutorial 2
August 9, 2018

1. Insert the numbers 39, 33, 26, 7, 14, 3, 71, 73, 67, 65 in this order in an initially empty R-B Tree. Draw the tree after each insertion. (Practice more such examples).

2. You are given with keys 10, 22, 31, 4, 15, 28, 17, 88, 59. You need to insert these keys into a hash table of length $m = 11$ using open addressing with the hash function $h(k) = k \bmod m$. Illustrate the result of inserting these keys using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k \bmod m$ and $h_2(k) = 1 + (k \bmod (m - 1))$. Now, repeat the same procedure for $m = 22$. (Practice more such examples).

3. Given a singly linked list $A_1, A_2, A_3, A_4,...,A_{n-2}, A_{n-1}, A_n$, design an algorithm to convert it to the singly linked list $A_1, A_n, A_2, A_{n-1}, A_3, A_{n-2},...$ and so on in O(n) time and O(1) space.

4. Given an array of n integers, design an O(n) time algorithm to find the next larger element for every element (next larger element of an element x is the first element that occurs after x in the array that is larger than x). (Hint: Use a stack).

5. In a binary tree, there is exactly one path between any two nodes (a path between two nodes informally is any sequence of edges that you can follow from one node to the other in the graphical representation of the tree). The path-sum of a path is the sum of the values of the nodes in the path. Design an O(n) time algorithm to find the maximum path-sum of any path in a given binary tree.

6. A Cartesian tree is a binary tree such that the value at any node x is greater than the value of any other node in the subtree rooted at x. Given an inorder traversal of a Cartesian tree, construct the tree. Is the tree unique? Justify your answer.

7. Given an array A of n integers such that there is at most k distinct values in A, design an O(nlog k) algorithm to sort the array. You can use at most O(k) space.

8. Given an array A of n integers an a positive integer k, design an algorithm to construct another array B such that $B[i] = \max\{A[j] \mid \max(i-k+1), 1) \leq j \leq i\}$ (i.e., B[i] is the sum of the last k elements of the array ending at i). Your algorithm should run in O(nlog k) time.

9. Given an array A of n integers, design an algorithm to find the total number of subarrays A[i..j] (part of A from index i to index j) such that $\sum A[k]$, $i \leq k \leq j$ is 0. Your algorithm should run in expected O(n) time.