Design and Analysis of Algorithms

Problem #1

You are given a list of unique numbers that were sorted in ascending order but then rotated. Consider the example list: [-9, -8, -6, -3, -15, -14].

- Write an $O(logn)$ algorithm to find the "rotation point" which is the index of the smallest number in the list given that list does not contain any duplicates. [*50 points*]

- Describe steps of your algorithm using the provided example list [*30 points*]; What is the final output of your algorithm (i.e., identifying the rotation point in the example list) [*20 points*]

Problem #2

Demonstrate the insertion of the keys 5, 28, 19, 15, 20, 33, 12, 17 into a hash table with collisions resolved by linear probing. Assume that the table has 9 slots and that the hash function is ***h(k) = k mod 9***. Draw the state of the hash table after every insertion [*60 Points*]. Propose strategies other than linear probing for handling collisions in a hash table [*20 Points*]. Are they better, worse? briefly explain [*20 Points*].

Problem #3

The input consists of a directed graph, encoded in the usual adjacency list representation, along with two distinct vertices *s* and *t*. Design and analyze a linear time algorithm to decide whether or not there is a path from *s* to *t* that contains a cycle.

Remember, your goal is to not only detect a cycle but to ensure that such a cycle can be part of a path starting from s and leading to t. (*100 Points*)