ECS 32B – Introduction to Data Structures
Homework 05

Due: November 6, 2020, 5:00pm PST

Important:

- The purpose of the homework is to practice **linear and binary search**. If you didn't use the corresponding searching technique as stated in the description, only half of the points will be given.
- You may upload as many times as you want before the deadline. Each time the autograder will let you know how many of the test cases you have passed/failed. To prevent people from manipulating the autogader, the content of only half of the test cases are visible.
- The testing files contain the test cases for which Gradescope shows the content. You may use the testing files to test your code locally before submitting to Gradescope. However, it does not contain the rest of the test cases, for which Gradescope does not show the content. So passing all the test cases in the testing files does not necessarily mean you will get 100.
- Please **do not copy others' answers**. Autograder can detect similar code.

Problem 1

Given a **sorted** (in ascending order) Python list of **distinct** integers, return all the indices i such that the elements at index i equals i. There may be more than one such index, so the return type is a Python list containing all indices that satisfy the criteria.

Write two functions for the same problem. The first function, `linearSearchValueIndexEqual`, uses linear search, and the second function, `binarySearchValueIndexEqual`, uses binary search. The test cases for the two functions in Gradescope are the same.

When the input list is empty, return an empty list.

Examples:

- Input: [0, 2, 5, 7]
  Output: [0]
  Explanation: The element at index 0 is 0, so the value equals its index.
- Input: [-5, 1, 2]
  Output: [1, 2]
  Explanation: The element at index 1 is 1, and the element at index 2 is 2.

Problem 2

Let `plist` be a Python list of distinct integers sorted in ascending order. Suppose `plist` is shifted to the right k times ($0 \leq k < len(plist)$). Determine the k, i.e., how many times the list is shifted to the right.

For example, `[1, 2, 3, 4, 5]` is sorted in ascending order. When it is shifted to the right once, the list becomes `[5, 1, 2, 3, 4]`, where every element moves to the right by one location and the last element, 5, wraps around to the beginning of the list.

Write two runctions for the same problem. The first function, `linearSearchShift`, uses linear search to find k, and the second function, `binarySearchShift`, uses binary search. The test cases for the two function in Gradescope are the same.

When the input is empty, return 0.

Examples:

- Input: [1, 2, 3, 4, 5]
  Output: 0
  Explanation: The list is not shifted so return 0. Note that we assume $0 \leq k < len(plist)$.
- Input: [5, 1, 2, 3, 4]
  Output: 1
- Input: [4, 5, 1, 2, 3]
  Output: 2
- Input: [3, 4, 5, 1, 2]
  Output: 3
- Input: [2, 3, 4, 5, 1]
  Output: 4