CS 372/469 – Spring 2022

Lab 1

Due: 02/20/2022 11:59 pm

For each of the following questions, write a successful running code in any programming language that you prefer. Your code should run without any errors for any *valid* input.

All problems are borrowed from Leetcode or GeeksForGeeks or TechieDelight.

Question 1 (25 points):

A peak element is an element that is strictly greater than its neighbors.
Given an integer array nums, find a peak element, and return its index. If the array contains multiple peaks, return the index to any of the peaks.
You may imagine that nums[-1] = nums[n] = -∞.
You must write an algorithm that runs in O(log n) time.
Source: https://leetcode.com/problems/find-peak-element/

Example 1:

```
Input: nums = [1,2,3,1]

Output: 2

Explanation: 3 is a peak element and your function should return the index number 2.
```

Example 2:

```
Input: nums = [1,2,1,3,5,6,4]

Output: 5

Explanation: Your function can return either index number 1 where the peak element is
2, or index number 5 where the peak element is 6.
```

Constraints:

- `1 <= nums.length <= 1000`
- $-2^{31} <= nums[i] <= 2^{31} - 1$

- `nums[i] != nums[i + 1]` for all valid `i`.

Question 2 (25 points):

Given a sorted array of non-negative distinct integers, find the smallest missing non-negative element in it.
Source (Open at your own risk – solution is also on this page):
https://www.techiedelight.com/find-smallest-missing-element-sorted-array/

For example,

Input: nums[] = [0, 1, 2, 6, 9, 11, 15]
Output: The smallest missing element is 3

Input: nums[] = [1, 2, 3, 4, 6, 9, 11, 15]
Output: The smallest missing element is 0

Input: nums[] = [0, 1, 2, 3, 4, 5, 6]
Output: The smallest missing element is 7

Question 3 (25 points):

Implement the Quicksort algorithm where the pivot is chosen randomly. The input to the problem should be an integer array and the output should be the sorted array.
Apart from what we learned in class, I found the following as the best and quickest explanation:
https://www.youtube.com/watch?v=Hoixgm4-P4M&ab_channel=MichaelSambol

Question 4 (25 points):

You are given two sorted lists of size m and n. Write an O(log m + log n) time algorithm for computing the kth largest element in the union of the two lists. The input to the problem are two sorted integer arrays and a target *k*, and the output is the kth largest element.
Source (Open at your own risk – solution is also on this page): https://www.geeksforgeeks.org/k-th-element-two-sorted-arrays/

```
Input : Array 1 - 2 3 6 7 9
```

```
        Array 2 - 1 4 8 10
        k = 5
Output : 6
Explanation: The final sorted array would be -
1, 2, 3, 4, 6, 7, 8, 9, 10
The 5th element of this array is 6.

Input : Array 1 - 100 112 256 349 770
        Array 2 - 72 86 113 119 265 445 892
        k = 7
Output : 256
Explanation: Final sorted array is -
72, 86, 100, 112, 113, 119, 256, 265, 349, 445, 770, 892
7th element of this array is 256.
```

Submission Instructions: Put all your solutions in a properly commented file named *hw1_lastname_firstname.EXTENSION*, where EXTENSION = the appropriate extension for the programming language that you chose.