Quiz Description

> Note:
> For every algorithm problem, you are required to **describe your algorithm clearly and easily to understand** firstly, then **write the pseudocode**, and the last step is **analyzing the complexity of time and space**.

Quiz 1

There are 4 problems in total. However, Problem 1 & 4 contain several multiple choice and fill in the blanks, which is not easy to recall exactly.
Fortunately, Problem 2 & 3 are both algorithm problems and more informative. Here is their description.

Problem 2 Rotate Array [30 points]

> Same as **LeetCode 189**, just change "rotate" to "exchange"

Given an integer array $nums$, rotate the array to the right by k steps, where k is non-negative.

```
Example:
Input: nums = [1, 2, 3, 4, 5, 6, 7], k = 3
Output: [5, 6, 7, 1, 2, 3, 4]
```

Problem 3 Kth Largest Element in an Array [20 points]

> Same as **LeetCode 215**

Given an integer array $nums$ and an integer k, return the kth largest element in the array.

Quiz 2

> ❤️❤️❤️ Good news!
> Quiz 2 has 120 points in total, and if you get 100+ points, the excess points will be added

> to Quiz 1! Also, you will have Problem 0 which has 50 points, and that means **as long as you write your name and student ID, you will get 50 points!**

Problem 1 Heap Building Time Complexity Proof [20 points]

The time complexity of turn sized-n array A into a binary heap on S via root-fix operator on dynamic array is $O(n)$, where A stores the values in set S.

Problem 2 Huffman Encoding [30 points]

Given (character, frequency) pairs as following:

H	N	S	O	E	Y	T	D
14	16	8	12	30	18	43	65

1. Show the detail steps of building its Huffman tree, i.e., draw the Huffman tree building process step by step.
2. Write down the corresponding scheme of the Huffman tree you obtained in (1), you only need to draw a table, which contains two columns, the left is the character, the right is its corresponding Huffman coding.
3. Write down the corresponding codes of string $HONESTY$.
4. Please proof that Huffman encoding is the optimum prefix code, i.e., the space cost is minimized.

Problem 3 Heap Sort [20 points]

A sorted array is created by repeatedly removing the root of the min-heap until the min-heap becomes empty. Given an array-based min-heap (as follows), fulfill the following table to show the elements in the min-heap during heap sort progress.

1	6	15	17	8	54	23	93	39	52	26	79
											null
										null	null
									null	null	null
								null	null	null	null

							null	null	null	null	null
						null	null	null	null	null	null
					null	null	null	null	null	null	null
				null	null	null	null	null	null	null	null
			null	null	null	null	null	null	null	null	null
		null	null	null	null	null	null	null	null	null	null
	null	null	null	null	null	null	null	null	null	null	null