Algorithms

Sorting Homework 3

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: LeetCode 581 - Shortest Unsorted Continuous Subarray

- Given an integer array nums, you need to find **one continuous subarray** that if you **only sort** this subarray in ascending order, then the **whole array** will be **sorted in ascending** order.
  - Return the shortest such subarray and output its length.
- int findUnsortedSubarray(vector<int> &nums)
- First: Develop O(nlogn) solution
- Second: Develop O(n) time and O(1) space solution
  - It requires analysis and **observation** skills
  - When we mention O(n): 99% of the time I don't mean count sort

Examples

Example 1:

```
Input: nums = [2,6,4,8,10,9,15]
Output: 5
Explanation: You need to sort [6, 4, 8, 10, 9] in ascending order to make the whole array sorted
in ascending order.
```

Example 2:

```
Input: nums = [1,2,3,4]
Output: 0
```

Example 3:

```
Input: nums = [1]
Output: 0
```

Problem #2: LeetCode 826 - Most Profit Assigning Work

You have `n` jobs and `m` workers. You are given three arrays: `difficulty`, `profit`, and `worker` where:

- `difficulty[i]` and `profit[i]` are the difficulty and the profit of the i^{th} job, and
- `worker[j]` is the ability of j^{th} worker (i.e., the j^{th} worker can only complete a job with difficulty at most `worker[j]`).

Every worker can be assigned **at most one job**, but one job can be **completed multiple times**.

- For example, if three workers attempt the same job that pays `$1`, then the total profit will be `$3`. If a worker cannot complete any job, their profit is `$0`.

Return the maximum profit we can achieve after assigning the workers to the jobs.

- int maxProfitAssignment(vector<int> &diff, vector<int> &pro, vector<int> &worker)
- We can do a brute force algorithm in O(NM). Can you do better?

Example 1:

```
Input: difficulty = [2,4,6,8,10], profit = [10,20,30,40,50], worker = [4,5,6,7]
Output: 100
Explanation: Workers are assigned jobs of difficulty [4,4,6,6] and they get a profit of [20,20,30,30] separately.
```

Example 2:

```
Input: difficulty = [85,47,57], profit = [24,66,99], worker = [40,25,25]
Output: 0
```

Problem #3: LeetCode 1887 - Reduction Operations to Make the Array Elements Equal

- *Note: Simplified problem statement*
- Given an integer array, your goal is to make all values **equal**.
- In each step you perform the following operations:
  - Find the largest value in nums (idx i).
  - Find the **next** largest value in nums **strictly** smaller than largest (idx j)
  - Reduce the large to small: nums[i] = nums[j]
- Return the **number of operations** to make all elements in nums equal.
- int reductionOperations(vector<int> &nums)
- Find O(nlogn) approach

Example 1:

```
Input: nums = [5,1,3]
Output: 3
Explanation: It takes 3 operations to make all elements in nums equal:
1. largest = 5 at index 0. nextLargest = 3. Reduce nums[0] to 3. nums = [3,1,3].
2. largest = 3 at index 0. nextLargest = 1. Reduce nums[0] to 1. nums = [1,1,3].
3. largest = 3 at index 2. nextLargest = 1. Reduce nums[2] to 1. nums = [1,1,1].
```

Example 2:

```
Input: nums = [1,1,1]
Output: 0
Explanation: All elements in nums are already equal.
```

Example 3:

```
Input: nums = [1,1,2,2,3]
Output: 4
Explanation: It takes 4 operations to make all elements in nums equal:
1. largest = 3 at index 4. nextLargest = 2. Reduce nums[4] to 2. nums = [1,1,2,2,2].
2. largest = 2 at index 2. nextLargest = 1. Reduce nums[2] to 1. nums = [1,1,1,2,2].
3. largest = 2 at index 3. nextLargest = 1. Reduce nums[3] to 1. nums = [1,1,1,1,2].
4. largest = 2 at index 4. nextLargest = 1. Reduce nums[4] to 1. nums = [1,1,1,1,1].
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."