Hacker Devils Easy Meeting

January 28, 2025

Meeting Agenda

- Announcements
- Officer Introductions
- Problems
- Pizza!

Attendance:

Join HKD Discord:



Problem 1: Best Time to Buy and Sell Stocks

https://leetcode.com/problems/best-time-to-buy-and-sell-stock/

Maximize your profit by choosing one day to buy the stock and then any day **AFTER** to sell the stock.
- given the price of the stock for each day in array prices

121. Best Time to Buy and Sell Stock

Solved

Easy | Topics | Companies

You are given an array `prices` where `prices[i]` is the price of a given stock on the `i^th` day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock.

Return *the maximum profit you can achieve from this transaction*. If you cannot achieve any profit, return `0`.

Example 1:

```
Input: prices = [7,1,5,3,6,4]
Output: 5
Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6),
profit = 6-1 = 5.
Note that buying on day 2 and selling on day 1 is not allowed
because you must buy before you sell.
```

Example 2:

```
Input: prices = [7,6,4,3,1]
Output: 0
Explanation: In this case, no transactions are done and the max
profit = 0.
```

Problem 1

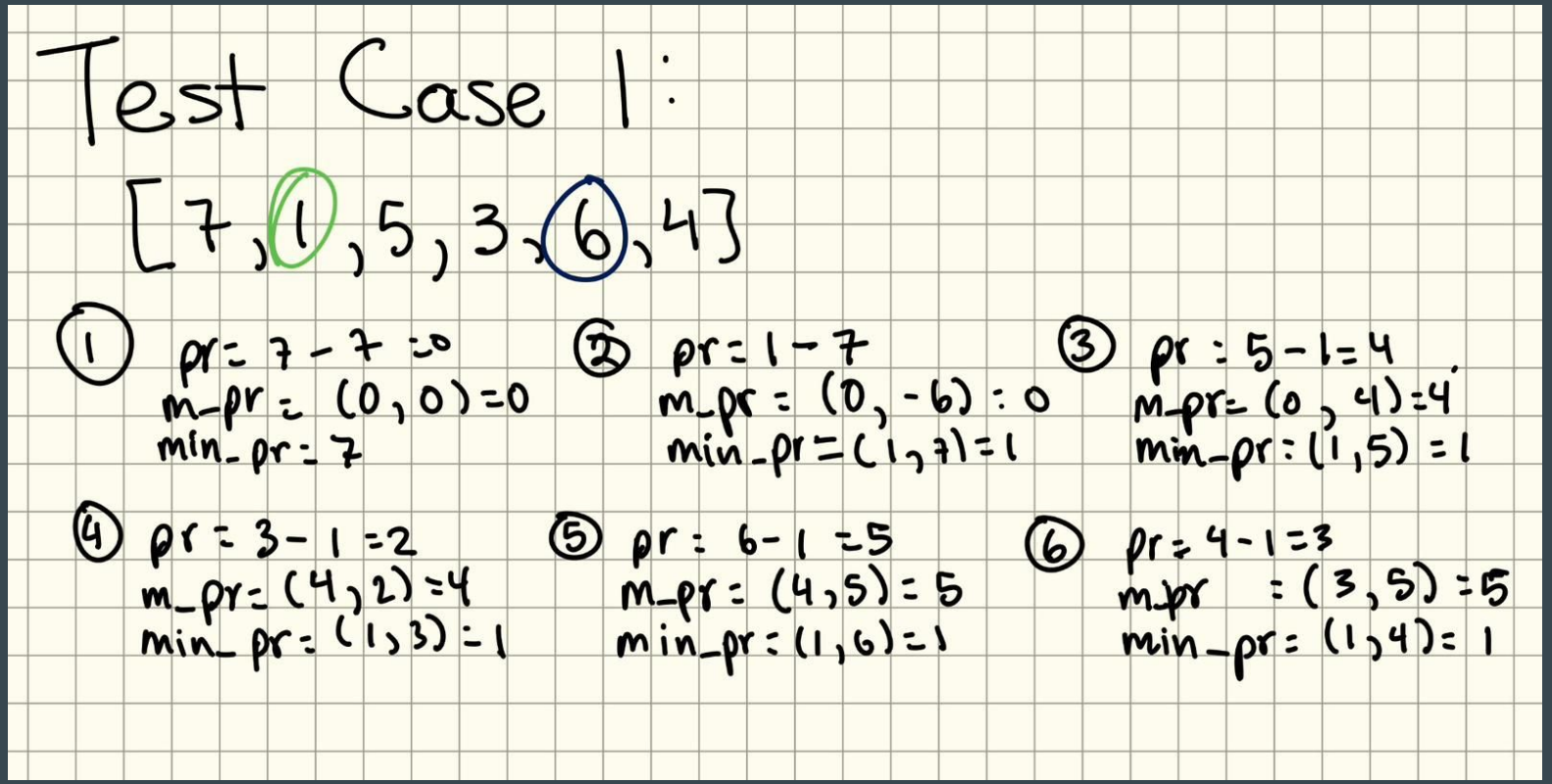
Give it a shot!

Problem 1: Solution Ideas

1. Since we are restricted to buying first and then selling:
    a. We can keep a best buy so far price (the lowest price you've seen up to the previous day)
    b. Calculate a temporary profit with that price and minimum price so far
    c. Compare the temporary profit and the maximum profit
    d. Compare the previous minimum price with the current price.

```
prices = [7,1,5,3,6,4]
```

Problem 1 - Test Case 1 Demo



Problem 1 Solution

```python
class Solution(object):
    def maxProfit(self, prices):
        """
        :type prices: List[int]
        :rtype: int
        """
        #best price to buy since ith day
        min_price = prices[0]
        max_profit = 0

        for i in range(len(prices)):
            #finding the profit from the minimum price and the selling price (the price on ith day)
            profit = prices[i]-min_price

            #comparing the profits and prices
            max_profit = max(profit, max_profit)
            min_price = min(prices[i], min_price)
        return max_profit
```

Problem 1 Solution - Java

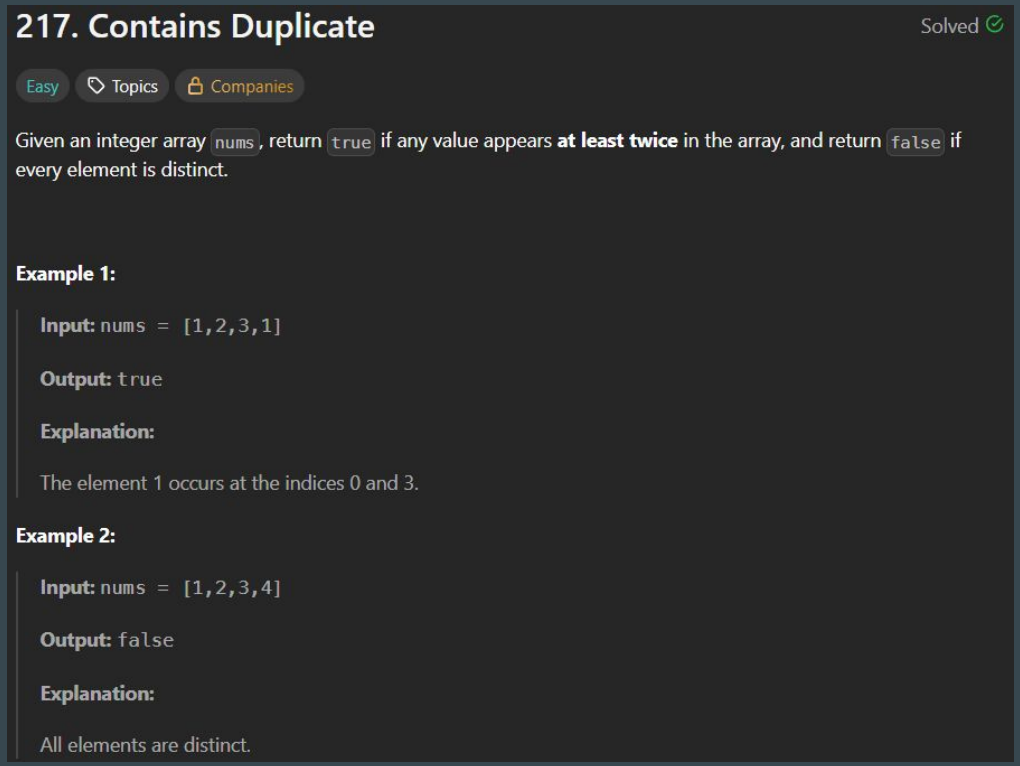
Java Auto

```java
class Solution {
    public int maxProfit(int[] prices) {
        int min_price = prices[0];
        int max_profit = 0;
        for(int i = 0; i < prices.length; i++){
            int profit = prices[i] - min_price;
            max_profit = Math.max(profit, max_profit);
            min_price = Math.min(prices[i], min_price);
        }
        return max_profit;
    }
}
```

Problem 2: Contains Duplicate

Given an array of integers, return **true** if there is a duplicate and **false** if there are no unique elements

https://leetcode.com/problems/contains-duplicate/description/



217. Contains Duplicate

Solved

Easy | Topics | Companies

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

Example 1:

> **Input:** `nums = [1,2,3,1]`
>
> **Output:** `true`
>
> **Explanation:**
>
> The element 1 occurs at the indices 0 and 3.

Example 2:

> **Input:** `nums = [1,2,3,4]`
>
> **Output:** `false`
>
> **Explanation:**
>
> All elements are distinct.

Problem 2: First Thought

Is there any way we can make finding a duplicate easier?

Problem 2

Give it a shot!

Problem 2: Solution: Brute Force

Initial Approach

- Compare every number with every other number
- Use nested for loops
- If two numbers are equal → **Duplicate** found

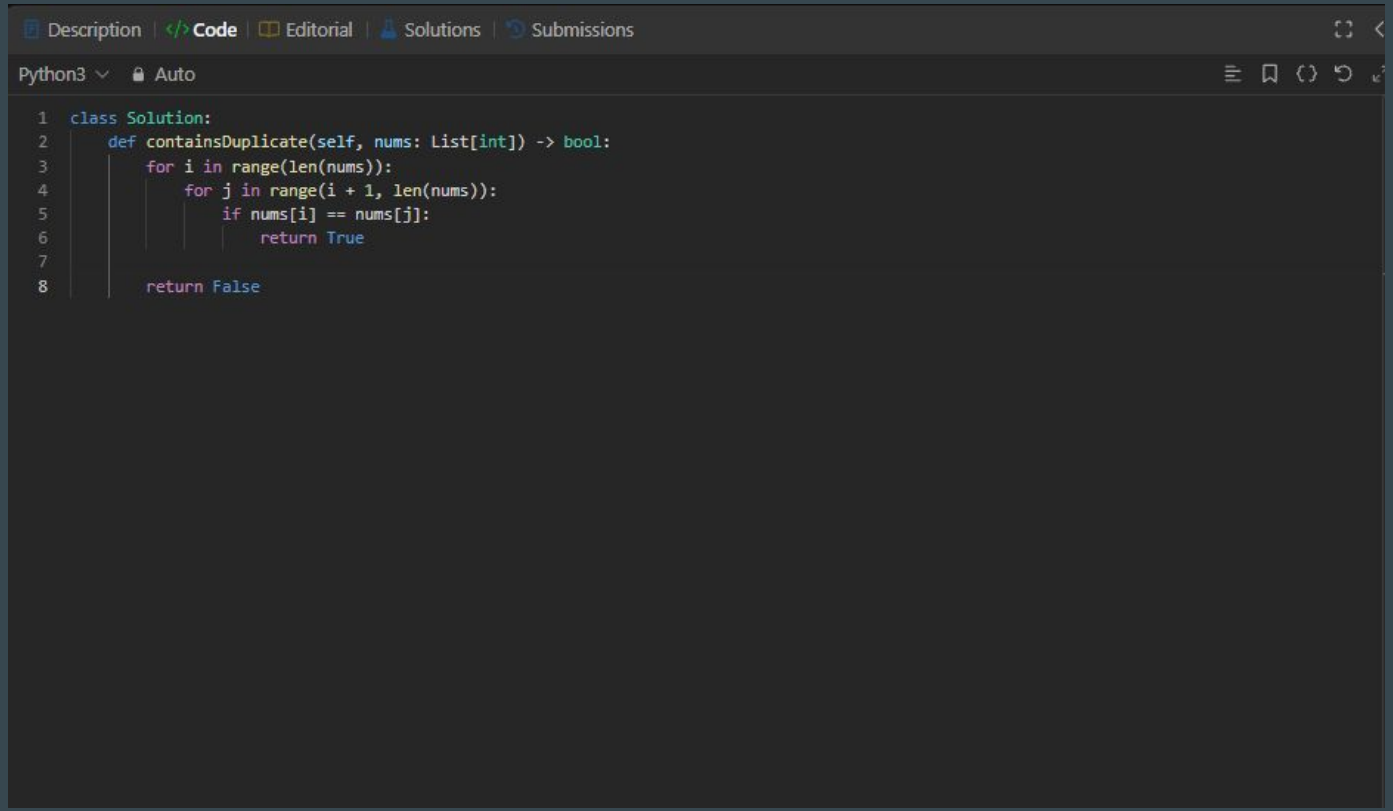
Why this makes sense

- Very intuitive
- No extra data structures

Problem 2: Solution

TRY CODING IT OUT!

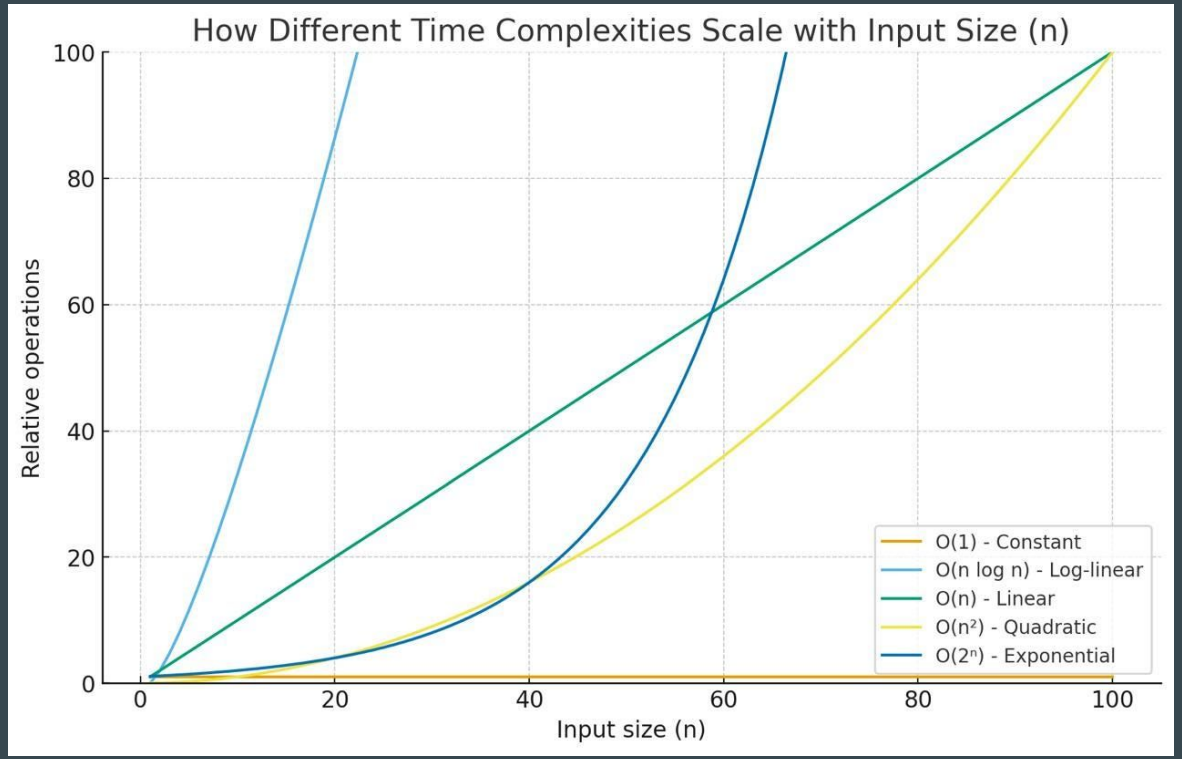
Problem 2 Solution:



Problem 2: It works but...

This solution works... but interviewers would not like this. This solution has a runtime of O(n^2).

It works for small inputs but as the input scales it becomes far too slow.



Problem 2: How can we optimize?

What data structure helps us track duplicates?

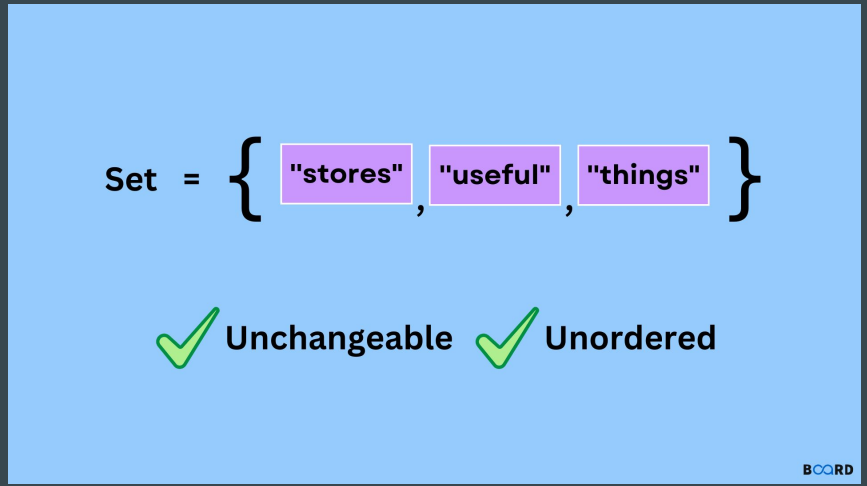
What if... we can track what we've already seen before.

Problem 2: Sets

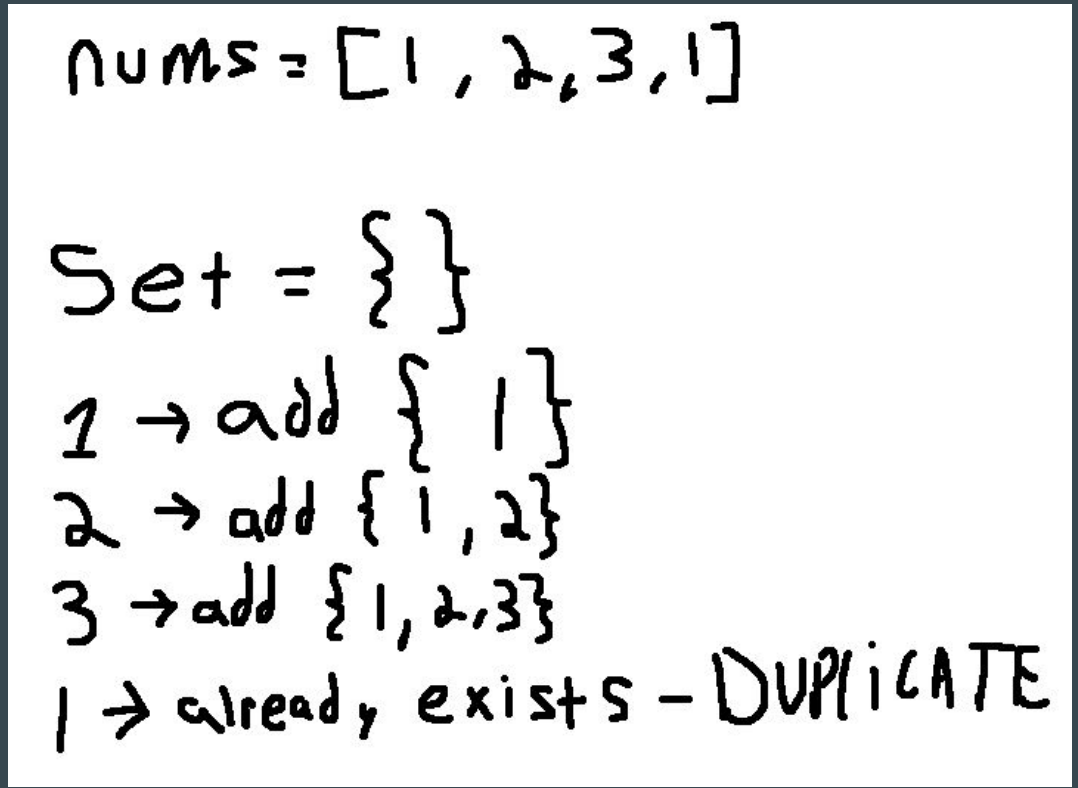
A set is an unordered collection of distinct elements. Think of it like a club membership roster where every member is listed only once.

Key Characteristics:

- **Uniqueness** : Each element in a set can appear only once. Adding a duplicate item has no effect [1, 2].
- **Unordered** : The elements in a set do not have a specific index or order (unlike an array or a list). You can check if an item exists in the set, but you can't access by index.
- **Efficient Operations** : Sets are highly optimized for fast lookups, insertions, and deletions.



Problem 2: Demo



Problem 2

Give it a shot!

Problem 2: Solution Continued

ython3 Auto

```python
class Solution:
    def containsDuplicate(self, nums: List[int]) -> bool:
        seen = set()

        for num in nums:
            if num in seen:
                return True

            seen.add(num)

        return False
```

Python

Java Auto

```java
class Solution {
    public boolean containsDuplicate(int[] nums) {
        Set<Integer> uniqueNumbers = new HashSet<Integer>();

        for (int num: nums) {
            if (uniqueNumbers.contains(num)) {
                return true;
            }
            uniqueNumbers.add(num);
        }

        return false;
    }
}
```

Java

PIZZA TIME!!!!