# Experiment 3.1

**Student Name: Yash Gupta** **UID: 20BCS5009**
**Branch: BE-CSE** **Section/Group:20BCS_DM-716 B**
**Semester: 6** **Date of Performance: 01/05/23**
**Subject Name: CC LAB** **Subject Code: 20CSP_351**

## 1. Aim:

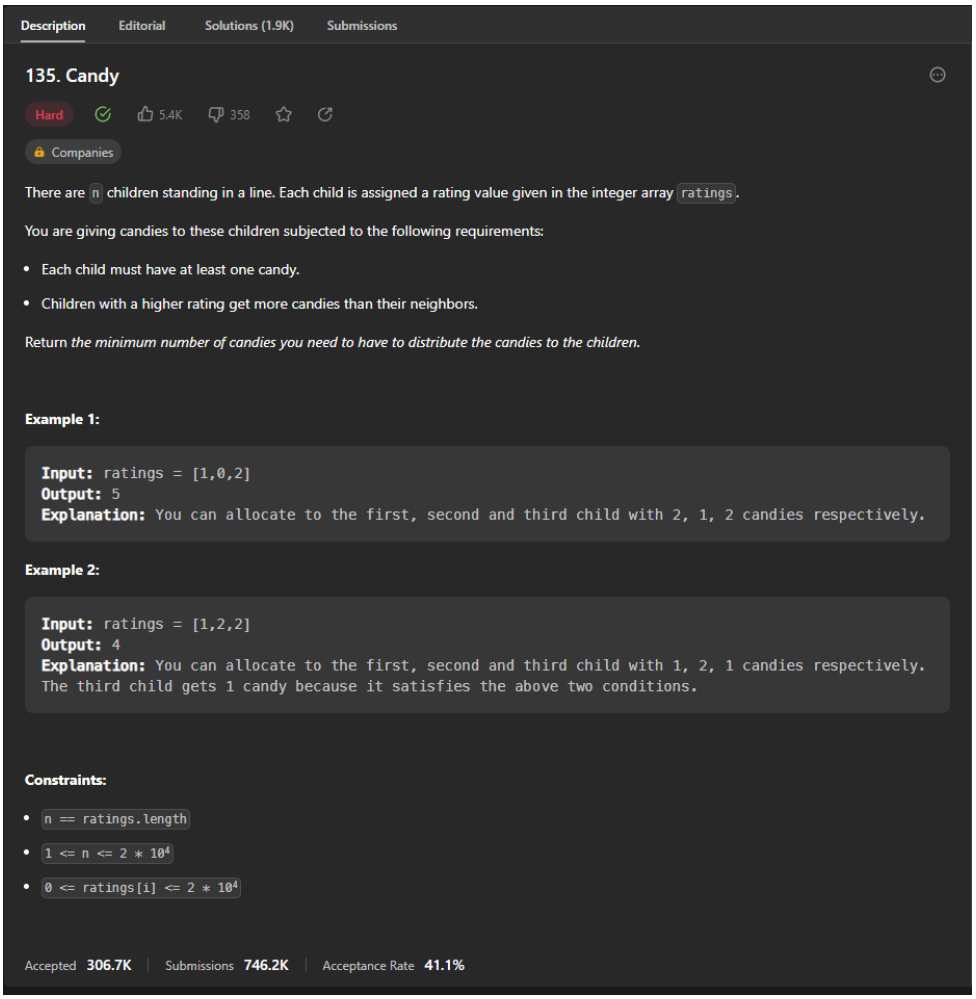
To implement the concept of Greedy Algorithm.

## 2. Objective:

- The objective is to build problem solving capability and to learn the basic concepts of data structures.
- Understand the problem and find out better approach to solve particular problem

## 3. LeetCode code and output:

### • Candy



Description Editorial Solutions (1.9K) Submissions

135. Candy

Hard 5.4K 358

Companies

There are `n` children standing in a line. Each child is assigned a rating value given in the integer array `ratings`.

You are giving candies to these children subjected to the following requirements:

- Each child must have at least one candy.
- Children with a higher rating get more candies than their neighbors.

Return *the minimum number of candies you need to have to distribute the candies to the children.*

**Example 1:**

```
Input: ratings = [1,0,2]
Output: 5
Explanation: You can allocate to the first, second and third child with 2, 1, 2 candies respectively.
```

**Example 2:**

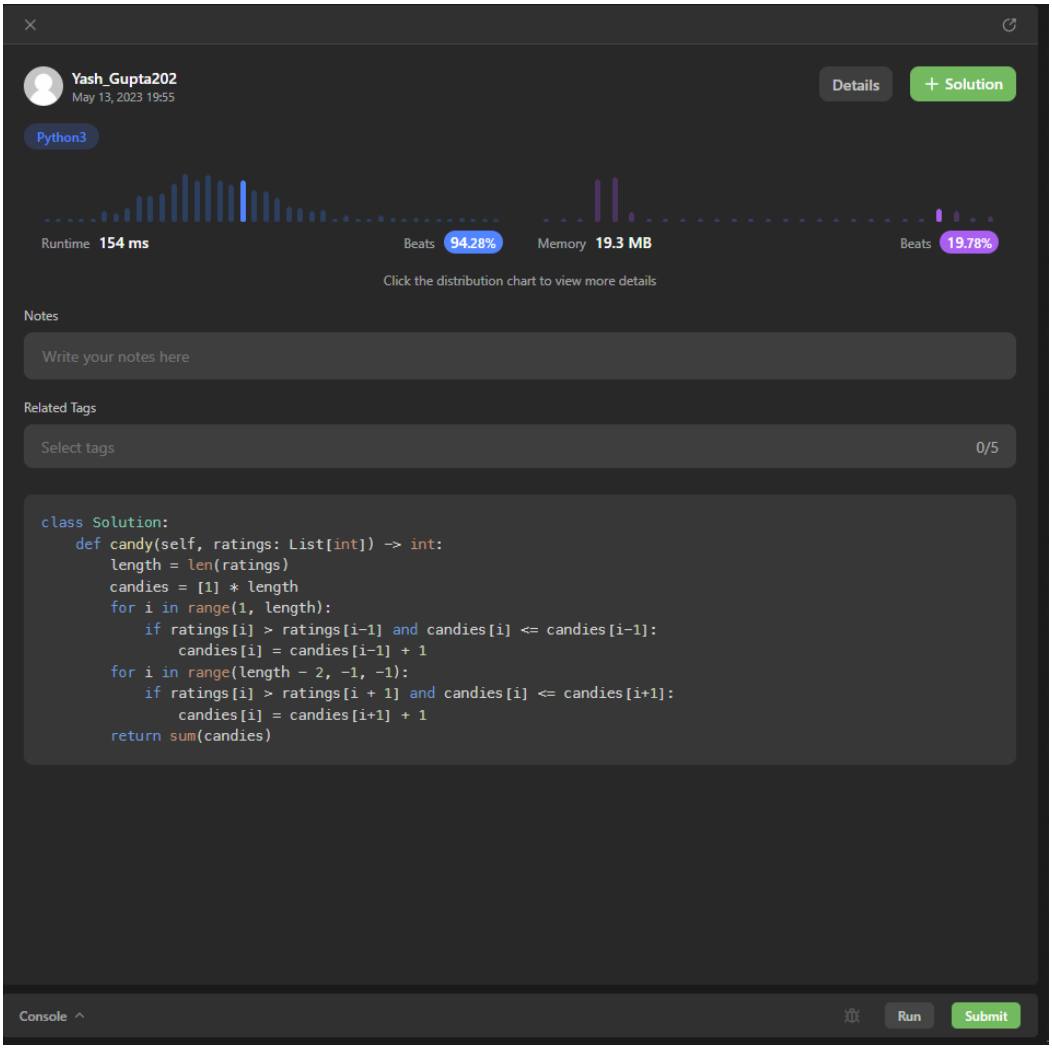
```
Input: ratings = [1,2,2]
Output: 4
Explanation: You can allocate to the first, second and third child with 1, 2, 1 candies respectively.
The third child gets 1 candy because it satisfies the above two conditions.
```

**Constraints:**

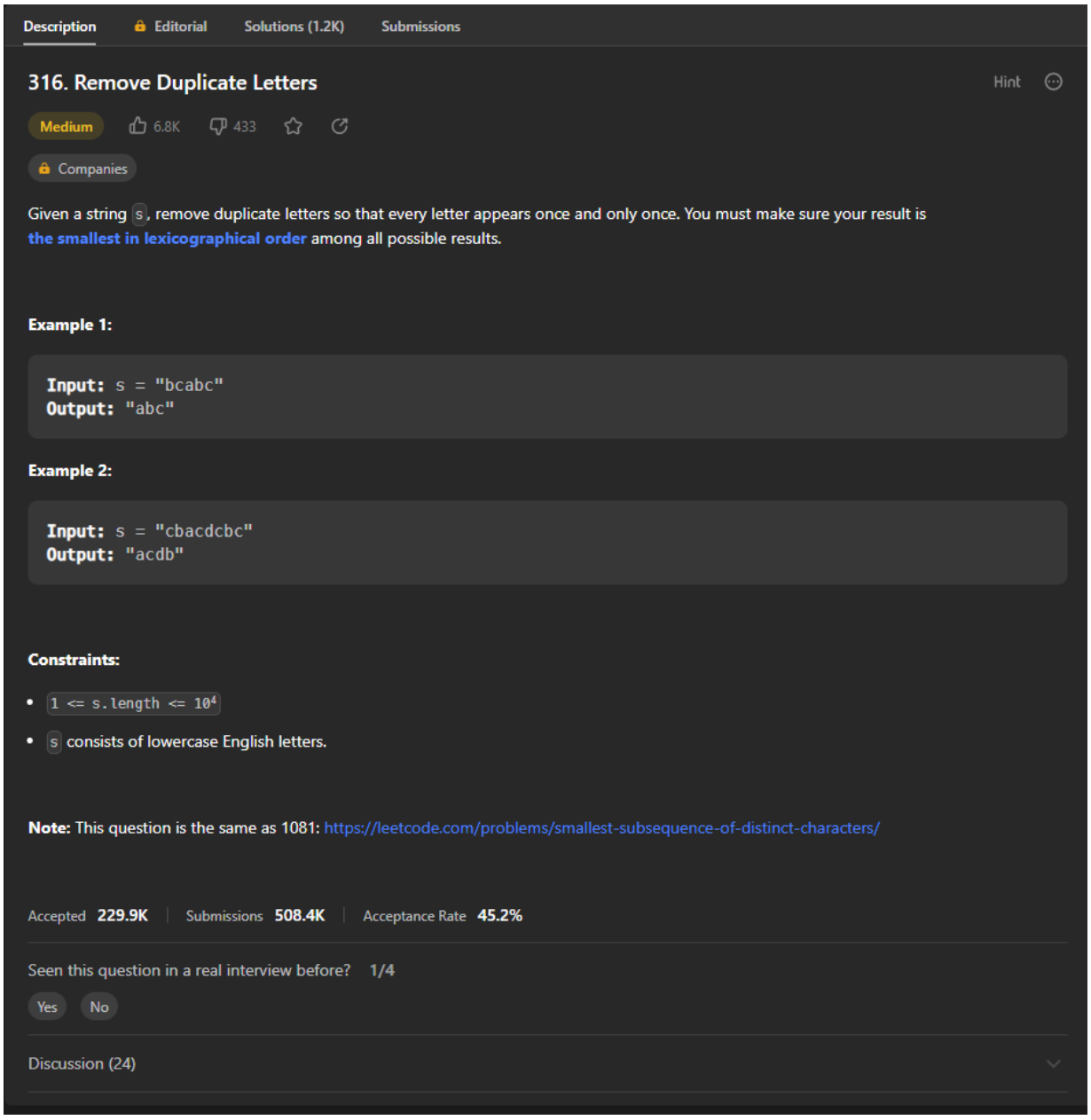
- `n == ratings.length`
- `1 <= n <= 2 * 10^4`
- `0 <= ratings[i] <= 2 * 10^4`

Accepted **306.7K** | Submissions **746.2K** | Acceptance Rate **41.1%**

```python
class Solution:
    def candy(self, ratings: List[int]) -> int:
        length = len(ratings)
        candies = [1] * length
        for i in range(1, length):
            if ratings[i] > ratings[i-1] and candies[i] <= candies[i-1]:
                candies[i] = candies[i-1] + 1
        for i in range(length - 2, -1, -1):
            if ratings[i] > ratings[i + 1] and candies[i] <= candies[i+1]:
                candies[i] = candies[i+1] + 1
        return sum(candies)
```



- **Remove Duplicate Letters**



Description | Editorial | Solutions (1.2K) | Submissions

316. Remove Duplicate Letters

Hint

Medium 6.8K 433

Companies

Given a string `s`, remove duplicate letters so that every letter appears once and only once. You must make sure your result is the smallest in lexicographical order among all possible results.

**Example 1:**

```
Input: s = "bcabc"
Output: "abc"
```

**Example 2:**

```
Input: s = "cbacdcbc"
Output: "acdb"
```

**Constraints:**

- `1 <= s.length <= 10^4`
- `s` consists of lowercase English letters.

**Note:** This question is the same as 1081: https://leetcode.com/problems/smallest-subsequence-of-distinct-characters/

Accepted 229.9K | Submissions 508.4K | Acceptance Rate 45.2%

Seen this question in a real interview before? 1/4

Yes No

Discussion (24)

```
class Solution:
    def removeDuplicateLetters(self, s: str) -> str:
        stack = []
        seen = set()
        last_occurance = {}
        for i in range(len(s)):
            last_occurance[ s[i] ] = i

        # print(last_occurance)

        for i, ch in enumerate(s):
            if( ch in seen ):
                continue
```

```
        else:
            # 3
            while( stack and stack[-1] > ch and last_occurance[stack[-1]] > i ):
                removed_char = stack.pop()
                seen.remove(removed_char)
            seen.add(ch)
            stack.append(ch)
        # print(stack)
    return ''.join(stack)
```