Candy Problem Explanation and Solution

The Candy problem is a common interview question that focuses on distributing candies to children such that:

1. Each child gets at least one candy.
2. A child with a higher rating than their neighbors gets more candies.

The task is to determine the minimum candies required to satisfy these conditions.

Approach Explanation

The solution is implemented using a two-pass algorithm:

1. **Left to Right Pass**:
   - Traverse the list from left to right. For each child, if their rating is higher than the previous child's rating, give them more candies than the previous child.

2. **Right to Left Pass**:
   - Traverse the list from right to left. For each child, if their rating is higher than the next child's rating, give them more candies than the next child (only if they don't already have more candies).

This ensures that all constraints are satisfied in both directions.

Time and Space Complexity

- **Time Complexity**: O(n), where n is the number of children. This is because we perform two linear passes through the ratings list.

- **Space Complexity**: O(n), for the array used to store the candy distribution.

Code Screenshot

```python
class Solution:
    def candy(self, ratings: List[int]) -> int:
        # 2 passes - one for left neighbors one for right neighbors

        arr = [1] * len(ratings)

        for i in range(1, len(ratings)):
            if ratings[i] > ratings[i-1]:
                arr[i] = arr[i-1] + 1

        for i in range(len(ratings) - 2, -1, -1):
            if ratings[i] > ratings[i+1]:
                if arr[i] <= arr[i+1]:
                    arr[i] = arr[i+1] + 1

        return sum(arr)
```