#238

Merge Intervals

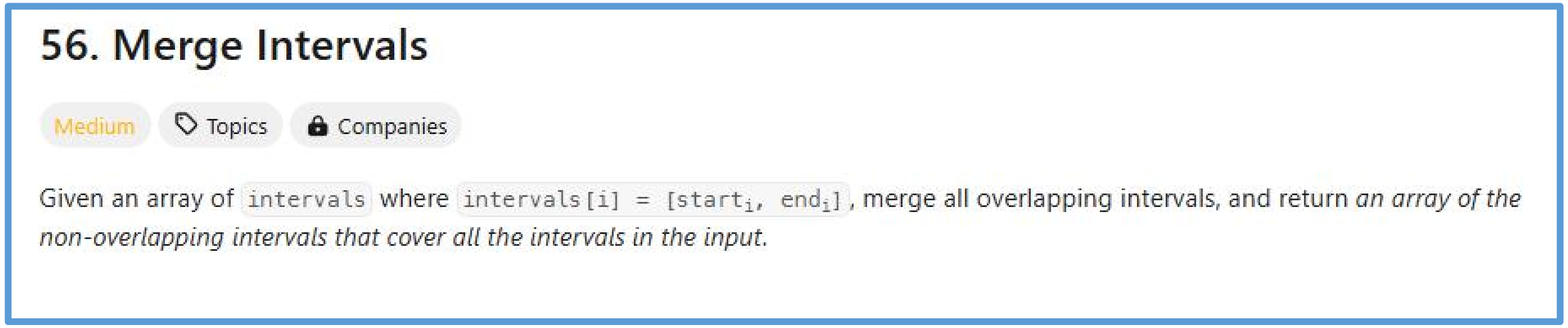
Akhtamov Azimjon **2024299010**

2025-01-09

Problem Definition (1)

- **Source**: Leetcode
- Title: **Merge Intervals**
- Difficulty: **medium**
- Type: **Arrays**

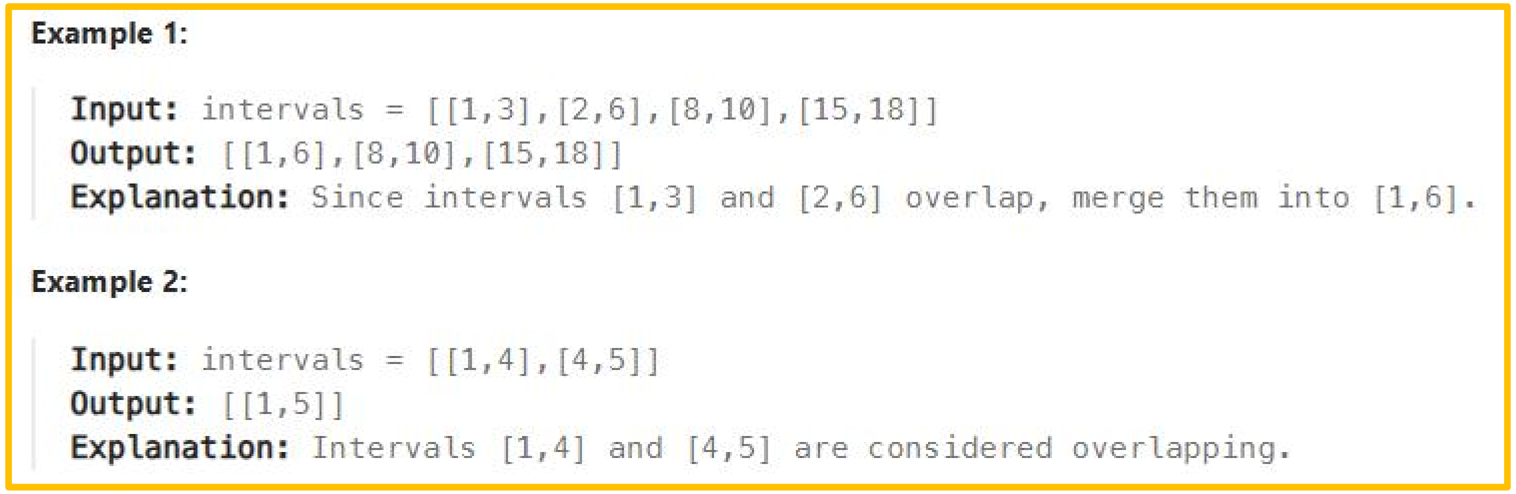
Problem Definition (1)



56. Merge Intervals

Medium · Topics · Companies

Given an array of `intervals` where $\texttt{intervals[i]} = [\text{start}_i, \text{end}_i]$, merge all overlapping intervals, and return *an array of the non-overlapping intervals that cover all the intervals in the input.*

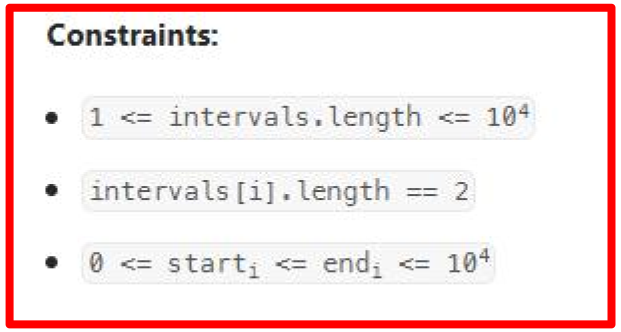


Example 1:

```
Input: intervals = [[1,3],[2,6],[8,10],[15,18]]
Output: [[1,6],[8,10],[15,18]]
Explanation: Since intervals [1,3] and [2,6] overlap, merge them into [1,6].
```

Example 2:

```
Input: intervals = [[1,4],[4,5]]
Output: [[1,5]]
Explanation: Intervals [1,4] and [4,5] are considered overlapping.
```



Constraints:

- $1 \le \texttt{intervals.length} \le 10^4$
- $\texttt{intervals[i].length} == 2$
- $0 \le \text{start}_i \le \text{end}_i \le 10^4$

Solution (1)

0 1	0 1	0 1	0 1
[1,3]	[2,6]	[7,9]	[8,10]
x	**y**		

x[0] <= y [0]

Overlap: **x[1] <= y [0]**

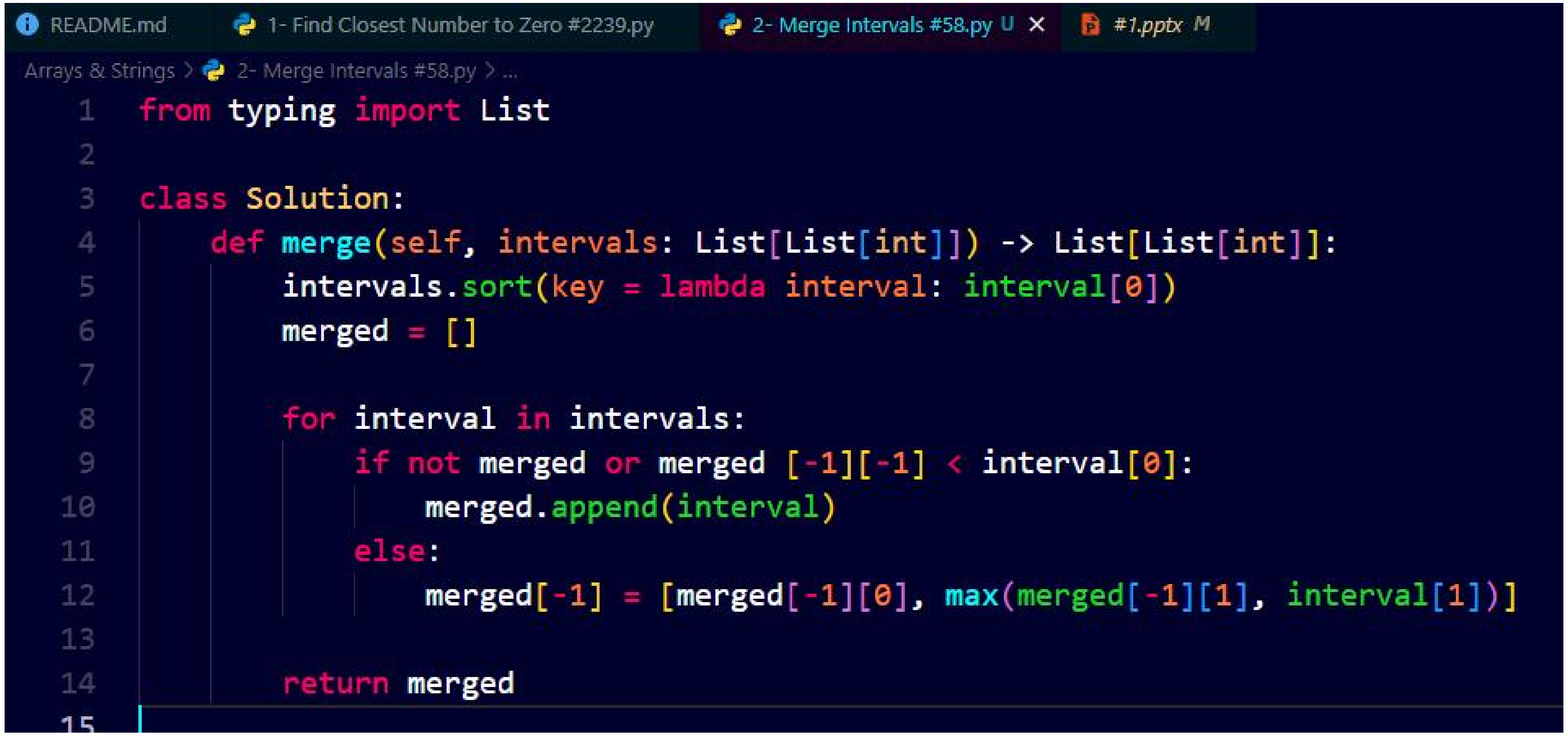
This case is **overlapping!**

merged

$\longleftrightarrow$

1 2 3 4 5 6 7 8 9 10

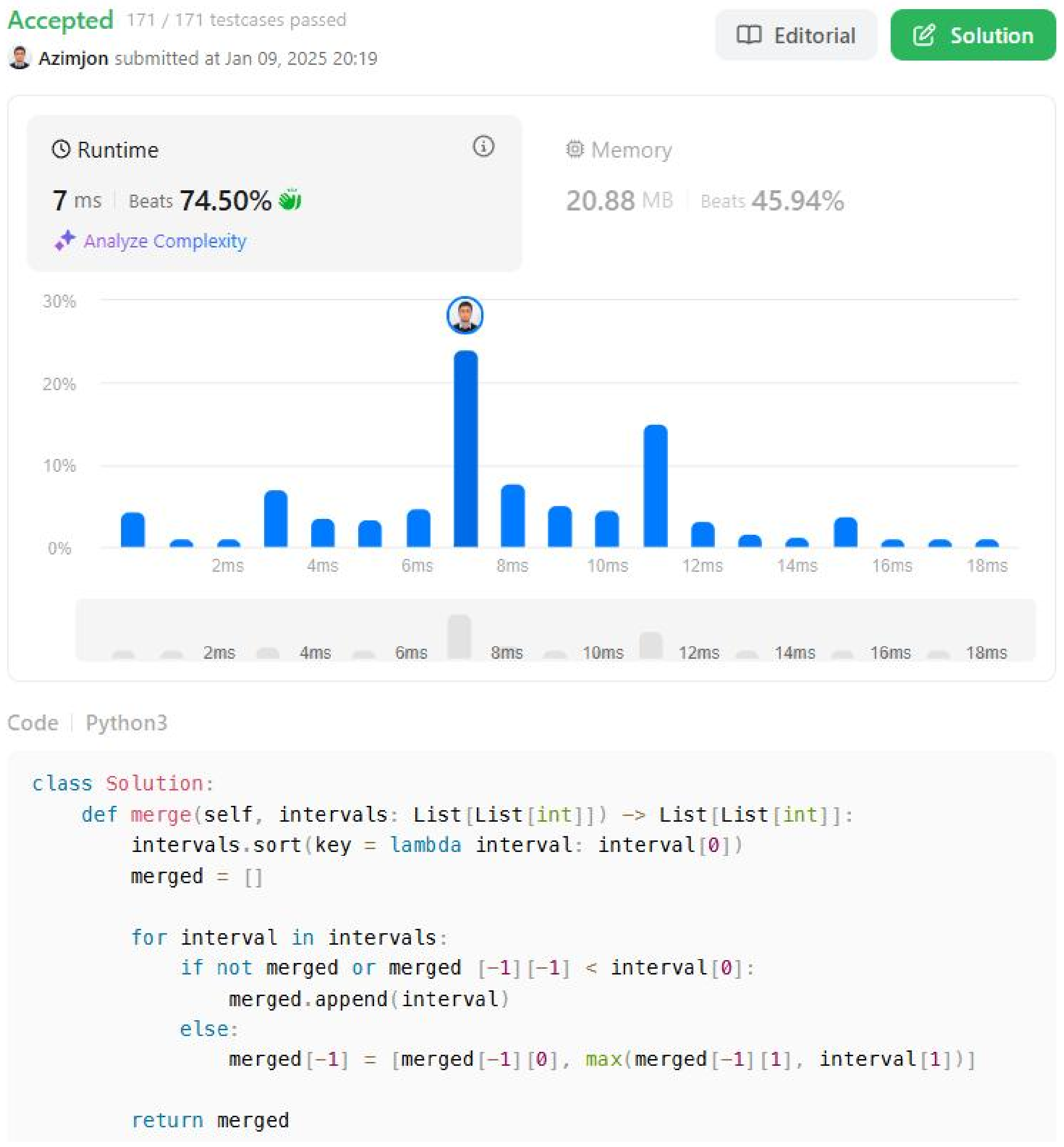
Solution (2)


README.md
1- Find Closest Number to Zero #2239.py
2- Merge Intervals #58.py U
#1.pptx M
Arrays & Strings > 2- Merge Intervals #58.py > ...

```python
from typing import List

class Solution:
    def merge(self, intervals: List[List[int]]) -> List[List[int]]:
        intervals.sort(key = lambda interval: interval[0])
        merged = []

        for interval in intervals:
            if not merged or merged [-1][-1] < interval[0]:
                merged.append(interval)
            else:
                merged[-1] = [merged[-1][0], max(merged[-1][1], interval[1])]

        return merged
```

lambda - to get any of them.
interval[0]: I started to sort from starting point
I merged all in a new list **merged = []** -> **merged[-1][-1]** is to check previus interval in **merged list**

Solution (3)


Accepted 171 / 171 testcases passed

Azimjon submitted at Jan 09, 2025 20:19

Runtime

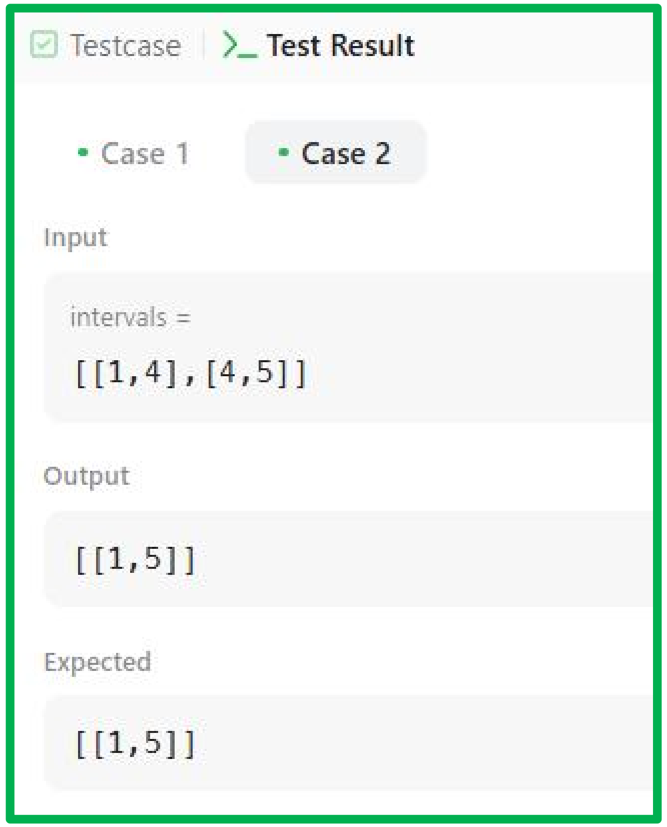
7 ms | Beats 74.50%

Memory

20.88 MB | Beats 45.94%

Code | Python3

```
class Solution:
    def merge(self, intervals: List[List[int]]) -> List[List[int]]:
        intervals.sort(key = lambda interval: interval[0])
        merged = []

        for interval in intervals:
            if not merged or merged [-1][-1] < interval[0]:
                merged.append(interval)
            else:
                merged[-1] = [merged[-1][0], max(merged[-1][1], interval[1])]

        return merged
```



Input

intervals =

```
[[1,4],[4,5]]
```

Output

```
[[1,5]]
```

Expected

```
[[1,5]]
```

What I have learned

❖ **Arrays:**

✓ I learnt to merge arrays without overlapping

✓ And I got how to sorting arrays on interval

✓ I used append() to store all selected intervals in a merged list

The merged list (merged = []) stores at most n intervals in the worst case.
so , this is total O(n).

Questions and Answers

Greetings