

Report: Combine Overlapping Ranges

Problem Statement:

Given an array of ranges where each range is represented as [begin, end], the goal is to combine all overlapping ranges and return a new array of non-overlapping ranges that cover all the input ranges.

Explanation:

1. Sorting the ranges:

- First, we sort the ranges based on their starting points. This helps us check for overlaps in order.

2. Merging logic: Start with the first range. For each following range, check if it overlaps with the current range.

-If they overlap, extend the current range to cover both.

-If they don't overlap, save the current range and move to the next.

3. Final result: The final list contains all non-overlapping ranges after merging all overlaps.

Python Code:

```
def merge_ranges(ranges):
    if not ranges:
        return []

    ranges.sort() # Sort ranges by start time
    merged = [ranges[0]]

    for start, end in ranges[1:]:
        last_end = merged[-1][1]

        if start <= last_end:
            merged[-1][1] = max(last_end, end)
        else:
            merged.append([start, end])

    return merged
```

Example1:

Input: $[[1,3],[2,6],[8,10],[15,18]]$

Output: $[[1,6],[8,10],[15,18]]$

Explanation: Ranges $[1,3]$ and $[2,6]$ overlap and are merged into $[1,6]$.

Example2:

Input: $[[1,4],[4,5]]$

Output: $[[1,5]]$

Explanation: Ranges $[1,4]$ and $[4,5]$ touch each other (overlap) and are merged into $[1,5]$.