

Combine Overlapping Ranges

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

Example Cases

Example 1:

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

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

Explanation: Ranges [1,3] and [2,6] overlap, so they are combined into [1,6].

المثال الأول

```
def merge_ranges(ranges):  
    # Step 1: Sort the ranges based on the start value  
    ranges.sort(key=lambda x: x[0])  
  
    # Step 2: Initialize the result list with the first range  
    merged = [ranges[0]]  
  
    # Step 3: Iterate through the sorted ranges  
    for current in ranges[1:]:  
        # Get the last range in the merged list  
        last = merged[-1]  
  
        # Check if the current range overlaps with the last range  
        if current[0] <= last[1]: # Overlap condition  
            # Merge the current range with the last range  
            merged[-1] = [last[0], max(last[1], current[1])]  
        else:  
            # No overlap, add the current range to the merged list  
            merged.append(current)  
  
    # Step 4: Return the merged ranges  
    return merged
```

Example 2:

Input: ranges = [[1,4],[4,5]]

Output: [[1,5]]

Explanation: Ranges [1,4] and [4,5] are considered overlapping.

1 Testing the Code

```
def merge_ranges(ranges):  
    # Step 1: Sort the ranges based on the start value  
    ranges.sort(key=lambda x: x[0])  
  
    # Step 2: Initialize the result list with the first range  
    merged = [ranges[0]]  
  
    # Step 3: Iterate through the sorted ranges  
    for current in ranges[1:]:  
        # Get the last range in the merged list  
        last = merged[-1]  
  
        # Check if the current range overlaps with the last range  
        if current[0] <= last[1]: # Overlap condition  
            # Merge the current range with the last range  
            merged[-1] = [last[0], max(last[1], current[1])]  
        else:  
            # No overlap, add the current range to the merged list  
            merged.append(current)  
  
    # Step 4: Return the merged ranges  
    return merged  
  
# Example 2  
ranges = [[1,4],[4,5]]  
print(merge_ranges(ranges)) # Output: [[1,5]]
```

1 Output

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