MERGE OVERLAPPING INTERVALS

☆ In this problem we are given an array of intervals and we are supposed to return the array containing the least amount of intervals where each overlapping one is merged.

Brute force solution is to sort the array of intervals and then merge adjacent intervals if first element of right one is less than the second element of the right one.

Pseudocode :

```
mergeIntervalsBrute(arr, N) {
    int ans[]
    for(i = 0 → N) {
        start = arr[i][0]
        end = arr[i][1]
        if (!ans.empty() && end <= ans[N-1][1]
            continue
        }
        for(j = i + 1 → N) {
            if(arr[j][0] <= end) {
                end = max(end, arr[j][1])
            } else break
        }
        ans.append({start, end})
}
```

```
    return ans
}
```

Time Complexity is O(N log N) for sorting and ~O(2N) for everything else. Space Complexity is O(N) is in worst case.

Optimal Solution also needs sorted array but does everything else in a single iteration.

Pseudocode :

```
mergeIntervals (arr, N) {
    int [] ans
    for (i = 0 → N) {
        if (ans.empty () || arr [i][0] > ans [N-1][1]) {
            ans.append(arr [i])
        } else {
            ans [N - 1][1] = max(ans [N - 1][1],
                                 arr [i][1])
        }
    }
    return ans
}
```