**[Merge Intervals](https://leetcode.com/problems/merge-intervals/)**

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** MergeIntervals {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[][] intervals = {{1, 3}, {2, 6}, {8, 10} , {15, 18}};

**int**[][] result = *merge*(intervals);

**for**(**int** i = 0 ; i < intervals.length ; i++) {

**for**(**int** j = 0 ; j < intervals[0].length ; j++) {

System.***out***.print(result[i][j] + " ");

}

System.***out***.println();

}

}

**public** **static** **int**[][] merge(**int**[][] intervals) {

**if**(intervals == **null** || intervals.length == 0 || intervals[0].length == 0) {

**return** **new** **int**[][] {};

}

Arrays.*sort*(intervals, (a, b) -> Integer.*compare*(a[0], b[0]));

List<**int**[]> result = **new** ArrayList<>();

**int**[] temp = intervals[0];

result.add(temp);

**for**(**int**[] interval : intervals) {

**if**(interval[0] <= temp[1]) {

temp[1] = Math.*max*(temp[1], interval[1]);

}

**else** {

temp = interval;

result.add(temp);

}

}

**return** result.toArray(**new** **int**[result.size()][]);

}

}

Time complexity : O(nlogn). n is no of elements in intervals

Space Complexity : O(n), n is no of elements in intervals