Problem Pattern

1. What are overlapping intervals?

Let’s say there are 2 intervals A(L1, R1) and B(L2, R2) where R1>=L1 and R2>=L2. The 2 intervals are overlapping if L2<=R1 as well as L2>=L1 or L1<=R2 as well as L1>=L2.

2. What does a merged interval look like after merging 2 overlapping intervals?
   A(L1, R1) and B(L2, R2)

Merged Interval M = (min(L1,L2), max(R1,R2))

3. Can you do it O(N) where N is the number of intervals.

Problem Approach

1. Sort the intervals on the basis of starting point and if 2 intervals have the same starting point sort them on the basis of ending point.
2. Now, select the first interval as the global starting and ending points.
3. Then traverse the list
4. if the start point of the interval is less than or equal to the global endpoint then update the global endpoint with the maximum of the current global endpoint and the endpoint of the current interval.
5. Else, the current interval doesn’t merge with the global interval. So, insert the global endpoints in the final answer and update the global endpoints with the current interval.

Problem Pseudocode

```
merge(intervals) {
     int n = intervals.size();
     for i=0 to n :
       v.insert({intervals[i][0],intervals[i][1]});
     sort(v);
     int ss=v[0].first;
     int se=v[0].second;
     for i=1 to n :
        if v[i].first is less than or equal to se:
        se=max(se,v[i].second);
        else :
           vector<int> t(2,0);
           t[0]=ss;t[1]=se;
           ans.insert(t);
           ss=v[i].first;
           se=v[i].second;

     vector<int> t(2,0);
     t[0]=ss;t[1]=se;
     ans.insert(t);
     return ans;
}
```