1288. Remove Covered Intervals

Given a list of `intervals`, remove all intervals that are covered by another interval in the list.

Interval `[a,b)` is covered by interval `[c,d)` if and only if `c <= a` and `b <= d`.

After doing so, return *the number of remaining intervals*.

Example 1:

Input: intervals = [[1,4],[3,6],[2,8]]
Output: 2
Explanation: Interval [3,6] is covered by [2,8], therefore it is removed.

Example 2:

Input: intervals = [[1,4],[2,3]]
Output: 1

Example 3:

Input: intervals = [[0,10],[5,12]]
Output: 2

Example 4:

Input: intervals = [[3,10],[4,10],[5,11]]
Output: 2

Example 5:

Input: intervals = [[1,2],[1,4],[3,4]]
Output: 1

Sort intervals in such an order that only previous ones are possible to cover current one.

1. Sort by the left bound, and when left bounds are equal, sort right bounds by reverse order;
2. Therefore, **no interval can cover previous ones**;
3. Loop through the `intervals`, whenever current right most bound < next interval's right bound, it means current interval can NOT cover next interval, update right most bound and increase counter by 1.

```python
def removeCoveredIntervals(self, intervals: List[List[int]]) -> int:
        intervals.sort(key=lambda x:(x[0],-x[1]))
        count = 0
        curr = 0
        for x,y in intervals:
            if curr<y:
                curr = y
                count = count+1
        return count
```