Problem #1288: Remove Covered Intervals (Medium)

<https://leetcode.com/problems/remove-covered-intervals/>

My Solution:

<https://leetcode.com/problems/remove-covered-intervals/discuss/879128/Simple-Python-3-Solution-Runtime-beats-98.55>

1. Sort the intervals so that the first component is in ascending order which is the primary sorting, and then the second component is in descending order which is the secondary sorting.
2. Iterate through the intervals list. We already have sorted the intervals so that the first component is in ascending order. So we only need to check the second component to see if the interval to the right is covered.
3. If so, it is removed.
4. If not, then proceed to the next interval in the list by incrementing the index.
5. Finally the list only contains intervals that cannot be covered by any other interval. Return the length of this list.

class Solution:

def removeCoveredIntervals(self, intervals: List[List[int]]) -> int:

intervals = sorted(intervals, key = lambda x : (x[0], -x[1]))

i = 0

while i < len(intervals) - 1:

if intervals[i][1] >= intervals[i+1][1]:

intervals.remove(intervals[i+1])

else:

i += 1

return(len(intervals))