Merge Intervals Proof

Simin Lin

May 21, 2019

1 Problem

Given a collection of time intervals, merge all overlapping intervals.

Example 1:

Input: [[1,3],[2,6],[8,19],[15,18]]
Output: [[1,6],[8,19]]
Explanation: Since intervals [1,3] and [2,6] overlaps, merge them into [1,6].
[15,18] is inside of [8,19], so merge them into [8,19].

Example 2:

Input: [[1,4],[4,5]]
Output: [[1,5]]
Explanation: Intervals [1,4] and [4,5] are considered overlapping.

Prove that sorting starting and end time separately and sorting them as paired will yield the same result regardless of the method used for merging intervals.

Example 1:

Sorting by the starting time as paired in the intervals in ascending order:
Starting: [1, 2, 8, 15]
Ending: [3, 6, 19, 18]
Sorting the starting and ending time sorted separately:
Starting: [1, 2, 8, 15]
Ending: [3, 6, 18, 19]
Prove that sorting starting and end time separately will yield the correct solution.
Output: [[1,6],[8,19]]

2 Proof by Strong Induction

Assume that the collection of time intervals is sorted by their starting time in ascending order.

If there are no overlaps among the time intervals, then sorting starting time and ending time separately will produce the same ordering of the elements in the starting and ending arrays as the paired arrays. Therefore, it suffices to prove that if there are overlaps among the time intervals, sorting starting and ending time separately will yield the same result as sorting them as paired regardless of the method used for merging intervals.

2.1 Base Case:

There are 2 intervals, requiring 1 merge.

Let the intervals be $[a_1, a_2]$ and $[a_3, a_4]$ where $a_2 \geq a_3$. We don't need to consider the case where $a_2 < a_3$ since there would be no overlap.

- $a_2 \geq a_3$ and $a_2 < a_4$
  - Sorting by the starting time as paired in the intervals
    - Starting: $[a_1, a_3]$
    - Ending: $[a_2, a_4]$
  - Sorting the starting and ending time separately
    - Starting: $[a_1, a_3]$
    - Ending: $[a_2, a_4]$
  - Since ordering of the elements in the starting and ending arrays in both cases is the same, sorting the starting and ending time will not affect the merge.
- $a_2 \geq a_3$ and $a_2 \geq a_4$
  - Sorting by the starting time as paired in the intervals
    - Starting: $[a_1, a_3]$
    - Ending: $[a_2, a_4]$
  - Sorting the starting and ending time separately
    - Starting: $[a_1, a_3]$
    - Ending: $[a_4, a_2]$
  - While the elements in the ending array are flipped, the merged interval will always be

    a_1 ———— a_3 ———— a_4 ———— a_2

    Therefore, regardless of the merging method (assuming the method is correct), the merged interval will remain as above.

2.2 Inductive Step:

Let k∈N and assume the claim holds true for n=k intervals. Let the intervals be

$$\big[[a_1, a_2], [a_3, a_4], ..., [a_{2k-1}, a_{2k}]\big]$$

Prove that the claim holds true for n=k+1 intervals. Let the intervals be

$$\big[[a_1, a_2], [a_3, a_4], ..., [a_{2k-1}, a_{2k}], [a_{2k+1}, a_{2k+2}]\big]$$

By the inductive hypothesis, we know that the claim holds for n=k, which means from $[a_1, a_2]$ to $[a_{a2k-1}, a_{a2k}]$, the starting and ending time can be sorted separately without affecting the merging result.
Therefore, we just need to show that adding $[a_{2k+1}, a_{2k+2}]$ to the collection won't affect the the merging result. Let m be a positive integer less than k. Let there be an array $[a_{2m+1}, a_{2m+2}]$ in the interval collection. Show that when $a_{2m+2} \geq a_{2k+1}$, the merging result will not be affected.
We don't need to consider $a_{2m+2} < a_{2k+1}$ since there would be no overlap.

- $a_{2m+2} \geq a_{2k+1}$ and $a_{2m+2} < a_{2k+2}$
  - Sorting by the starting time as paired in the intervals
    - Starting: $[..., a_{2m+1}, ..., a_{2k+1}]$
    - Ending: $[..., a_{2m+2}, ..., a_{2k+2}]$
  - Sorting the starting and ending time separately
    - Starting: $[..., a_{2m+1}, ..., a_{2k+1}]$
    - Ending: $[..., a_{2m+2}, ..., a_{2k+2}]$
  - Since ordering of the elements in the starting and ending arrays in both cases is the same, sorting the starting and ending time will not affect the merge.
- $a_{2m+2} \geq a_{2k+1}$ and $a_{2m+2} \geq a_{2k+2}$
  - Sorting by the starting time as paired in the intervals
    - Starting: $[..., a_{2m+1}, ..., a_{2k+1}]$
    - Ending: $[..., a_{2m+2}, ..., a_{2k+2}]$
  - Sorting the starting and ending time separately
    - Starting: $[..., a_{2m+1}, ..., a_{2k+1}]$
    - Ending: $[..., a_{2k+2}, ..., a_{2m+2}]$
  - In this case, we know that $[a_{2m+1}, a_{2m+2}]$ and $[a_{2k+1}, a_{2k+2}]$ will always merge into 1 interval $[a_{2m+1}, a_{2m+2}]$:

    a_{2m+1}——...——a_{2k+1}——...——a_{2k+2}——...——a_{2m+2}

    Therefore, regardless of the merging method (assuming the method is correct), the merged interval will remain as above.

2.3 Conclusion:

since m is a positive integer less than k and we have proven that merging $[a_{2m+1}, a_{2m+2}]$ and $[a_{2k+1}, a_{2k+2}]$ will not affect the merging result, we have proven that the claim holds true ∀n∈N by the principle of strong induction.

References