

Analyze worst case time complexity of above algorithm :

Sorting intervals according to the right - most ends :

Implement "Quick Sort", and the worst case time complexity will be

$O(n^2)$ , which occurs when the selected pivot is always an extreme value.

(The smallest or largest)

Doing the interval selection :

Even in the worst case (going through all of the intervals), the time complexity will still in linear time  $\Rightarrow O(n)$

---

$$\Rightarrow n^2 + n$$

$$\Rightarrow n^2$$

$$\Rightarrow O(n^2) \text{ For worst case time complexity.}$$

Week 8

410821316

鄭復文

\*

Analyze worst case time complexity of above algorithm with sorted  $V$  :

We don't need to execute the sorting progress, then the only thing that can matter the time complexity is the interval selection progress. For interval selection progress, even in the worst condition (going through all of the ~~conditions~~ intervals), the time complexity will still in linear time. Thus

$$\Rightarrow O(n)$$

#

Rewrite above algorithm to detect target range may impossible to be covered by original interval set. Analyze space complexity, time complexity of best case and worst case of modified algorithm:

1. Set current position  $p \rightarrow l^*$ , answer set  $\mathcal{V}' \leftarrow \{\}$
2. Sort the intervals using quick sort
3. Check if the first interval are able to cover the starting end
4. Find right-most (have largest right end) interval  $v \in \mathcal{V}$  that  $l_v \leq p$
5. Update  $p \leftarrow r_v$ ,  $\mathcal{V}' \leftarrow \mathcal{V}' \cup \{v\}$
6. Check the interval is combined together or overlap with each other. If not, break and return (-1)
7. Repeat the steps until  $p \geq r^*$

Best Case Time Complexity:  $O(n \log n)$  (with sorting)

Best Case Space Complexity:  $O(n)$

Worst Case Time Complexity:  $O(n^2)$  (without sorting)

Worst Case Space Complexity:  $O(n)$

Above algorithm is doing through left to right.

Please rewrite it to a through right to left version. Is your new algorithm always get the same result of original one? Give examples to explain your answer:

In right to left:

1. Set current position  $l^* \rightarrow p$ ,  
answer set  $\{\} \rightarrow \mathcal{V}'$
2. Find left-most (have largest left end) interval  
 $\mathcal{V} \in \mathcal{V}$  that  $p \geq l_{\mathcal{V}}$
3. Update  $\mathcal{V}' \cup \{\mathcal{V}\} \rightarrow \mathcal{V}'$ ,  $r_{\mathcal{V}} \rightarrow p$
4. Repeat step 3 and step 2 until  $r^* \leq p$

\*

---

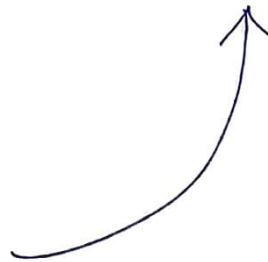
The new algorithm always gets the same result of the original one:

Left to Right :  $p \leftarrow 3+2$

Result :  $p = 5$

Right to Left :  $3+2 \rightarrow p$

Result :  $p = 5$



\*