

Insert Interval

[My Submissions \(/problems/insert-interval/submissions/\)](/problems/insert-interval/submissions/)

Total

Question

Solution

Accepted:

41387 Total Submissions: **189774** Difficulty: **Hard**

Given a set of *non-overlapping* intervals, insert a new interval into the intervals (merge if necessary).

You may assume that the intervals were initially sorted according to their start times.

Example 1:

Given intervals [1,3],[6,9] , insert and merge [2,5] in as [1,5],[6,9] .

Example 2:

Given [1,2],[3,5],[6,7],[8,10],[12,16] , insert and merge [4,9] in as [1,2],[3,10],[12,16] .

This is because the new interval [4,9] overlaps with [3,5],[6,7],[8,10] .

[Show Tags](#)[Show Similar Problems](#)Have you met this question in a real interview? [Discuss \(/discuss/questions/oj/insert-interval\)](/discuss/questions/oj/insert-interval)

Python



```
1 # Definition for an interval.
2 # class Interval(object):
3 #     def __init__(self, s=0, e=0):
4 #         self.start = s
5 #         self.end = e
6
7 class Solution(object):
8     def insert(self, intervals, newInterval):
9         """
10         :type intervals: List[Interval]
11         :type newInterval: Interval
12         :rtype: List[Interval]
13         """
14         i=0; length=len(intervals); top=0
15         while i<length:
16             if newInterval.start < intervals[i].start:
17                 intervals.insert(i, newInterval)
18                 return intervals
19             elif intervals[i].end < newInterval.start:
20                 top=i
21             elif intervals[i].start < newInterval.start < intervals[i].end:
22                 intervals[i].start = newInterval.start
23             elif intervals[i].start < newInterval.start < intervals[i].end:
24                 intervals[i].end = newInterval.end
25             i+=1
26         intervals.insert(top, newInterval)
27         return intervals
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
16     if length==0:
17         returnColumn.append(newInterval)
18         return returnColumn
19     while i<length:
20         if intervals[i].start >= newInterval.start:
21             if top==0 or returnColumn[top-1].end < newInterval.start:
22                 returnColumn.append(newInterval)
23                 top+=1
24             else:
25                 returnColumn[top-1].end = max(returnColumn[top-1].end,newInterval.end)
26             break
27         returnColumn.append(intervals[i])
28         top+=1
29         i+=1
30     while i<length:
31         if returnColumn[top-1].end >= intervals[i].start:
32             returnColumn[top-1].end = max(returnColumn[top-1].end,intervals[i].end)
33         else:
34             returnColumn.append(intervals[i])
35             top+=1
36         i+=1
37     if returnColumn[top-1].start < newInterval.start:
38         if returnColumn[top-1].end >= newInterval.start:
39             returnColumn[top-1].end = max(returnColumn[top-1].end,newInterval.end)
40         else:
41             returnColumn.append(newInterval)
42         top+=1
```

Custom Testcase ☐

Run Code

Submit Solution

Submission Result: Accepted (/submissions/detail/40365636/)

More Details > (/submissions/detail/40365636/)

Next challenges: (M) Two Sum II - Input array is sorted (/problems/two-sum-ii-input-array-is-sorted/)

(M) Word Search (/problems/word-search/)

(E) Rotate Array (/problems/rotate-array/)

Share your acceptance!

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy

Copyright © 2015 LeetCode