

HW7-Interval Search Tree

2024.4.29

112-2 PDSA HW7

Release date: 04/29 16:00

TA hour: 05/06 13:30

Due date: 05/08 21:00

Description:

An **Interval Search Tree (IST)** is an advanced and dynamic data structure that efficiently manages a collection of closed intervals on an one-dimensional numerical line. These intervals are denoted as $[a, b]$, where a and b represent the starting and ending points, respectively, and it is guaranteed that $(a \leq b)$ for each interval.

This data structure is optimized for speed due to its clever organization. Each node in the tree stores an interval with its ending points and the maximum ending point in its subtree, allowing for quickly discarding non-relevant intervals during searching. This design allows for basic operations—insertion, deletion, and overlap searching—to be performed in logarithmic time relative to the number of intervals. Here's how an IST operates:

- **Insertion:** An interval is added to the tree based on its starting point. If starting points of two intervals are the same, the ending points are compared. If an interval is identical to an existing node, then the value of that node is updated accordingly.
- **Deletion:** If an interval to be deleted exists, it should be deleted after the call, and the tree will restructure itself accordingly. If there are no intervals eligible for removal, no action will be taken.
- **Overlap Searching:** The tree efficiently finds all intervals that overlap with a given interval by comparing endpoints. Values of the overlapping intervals are returned.

Hint

- [Interval Search Tree Tutorial](#)

TestCase

Time Limit: 20 ms

N = Number of function call

M = Maximum time

Case:

case1: 20 points, $N \leq 6$, $M \leq 25$

case2: 20 points, $N \leq 10$, $M \leq 25$

case3: 20 points, $N \leq 200$, $M \leq 25$

case4: 20 points, $N \leq 300$, $M \leq 50$

case5: 20 points, $N \leq 500$, $M \leq 75$

