

# Lab1 – Interval Tree Implementation

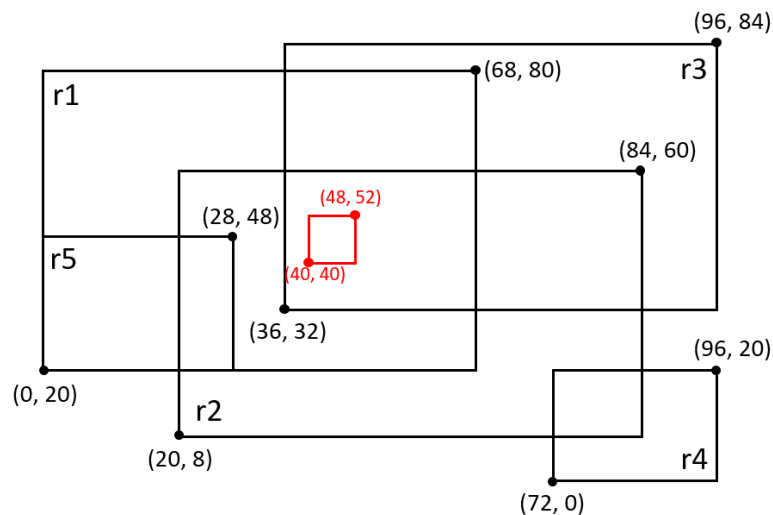
Deadline: 2021/11/12 23:55

## Lab1 Introduction

This is an exercise lab to review augmenting data structures.

You have to implement **interval tree** in this lab. VLSI databases commonly represent an integrated circuit as a list of rectangles. Assume that each rectangle is rectilinearly oriented (sides parallel to the x- and y-axes), so that we represent a rectangle by its minimum and maximum x and y-coordinates.

## Input



## Example (input.txt)

```
I r1 0 20 68 80 // action mode, rectangle ID, coordinate of lower-left corner, coordinate of upper-right corner
I r2 20 8 84 60
I r3 36 32 96 84
I r4 72 0 96 20
I r5 0 20 28 48
O 40 40 48 52
S 20 8 84 60
D 20 8 84 60
A 0 20
```

## Explanation of the Input

- Each line defines the action mode, rectangle ID, coordinate of lower-left corner and coordinate of upper-right corners.
- action mode
  - I: Insert the rectangle
  - D: Delete the rectangle
  - O: Overlap detection
  - S: Search the rectangle
  - A: Area. Find the rectangle which area is smallest with the same given coordinate of lower-left corner rectangle.

Note: All the coordinate values in input file are integer.

## Output

### Output Format (output.txt)

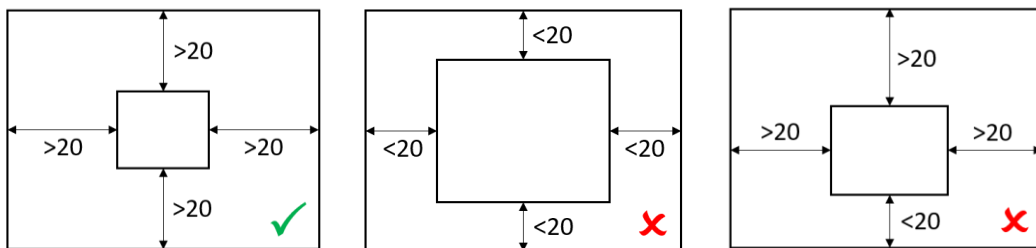
```
O           // action mode
r1          // the rectangle which cover the pattern rectangle
r2 violate
r3 violate
S           // action mode
r2          // the rectangle ID
A
r5
```

### Explanation of the Output

1. Only output when the action mode is O, S and A.
2. In each action, the first line output the action mode.
3. If the mode is O, please output each rectangle ID that the rectangle overlaps with the pattern rectangle and output “**violate**” if one of the edge is smaller than the distance constraint. Only output the rectangle which cover the pattern rectangle. Please output the rectangle ID in ascending order.

**Distance constraint is set as 20.**

For example:



4. If the mode is S, please output the rectangle ID.
5. If the mode is A, please output the rectangle ID which area is smallest with the same given coordinate of lower-left corner rectangles.
6. Please output the action mode in the input file in order.

## Environment

1. Linux (Please make sure your code is available on our linux server. If it cannot be executed, .zip file, you will get zero point!!)
2. Please use `argc` and `argv` to read input and output files or you will get fail in this lab.

## Submission

Please upload the following materials in a .zip file (e.g. **<student\_ID>.zip**) to New E3 by the deadline, specifying your student ID in the subject field. (If your submission file is not .zip file, you will get zero point!!)

1. Source code (.cpp) (only one!!)
2. Report

### Evaluation

1. You **MUST WRITE YOUR OWN CODE**. Copying codes may get **FAIL** in this course.
2. For each case, it will be regarded as “failed” if you use more than time limit.
3. Naming rule.
  - A. Compile procedure: **g++ -std=C++11 <student\_ID>.cpp -o Lab1**
  - B. Execution procedure: **./Lab1 [input] [output]**
  - C. Source code: **<student\_ID>.cpp**
  - D. Report: **<student\_ID>\_report.pdf**
  - E. .zip file: **<student\_ID>.zip** (compress your source code and report)
  - F. Not following specified naming rule will receive zero mark
4. Grading

A.	Small case(x3)	60%
	(Time limit: 1 minute for each small case)	
B.	Big case	30%
	Correct answer	10%
	Timing performance (if the answer is correct)	20%
	(Time limit: 5 minutes for each big case)	
C.	Report	10%
	• No more than 2 page	
	• Your report must contain:	
	i. Time complexity	
	ii. The flow chart of you program	

### Due date

- Due date : 2021/11/12 23:55
- **Penalty of 10%** of the total score per day for the first four days (weekend included) and will not be accepted afterwards