#### Segment Trees

HKOI Training 2012

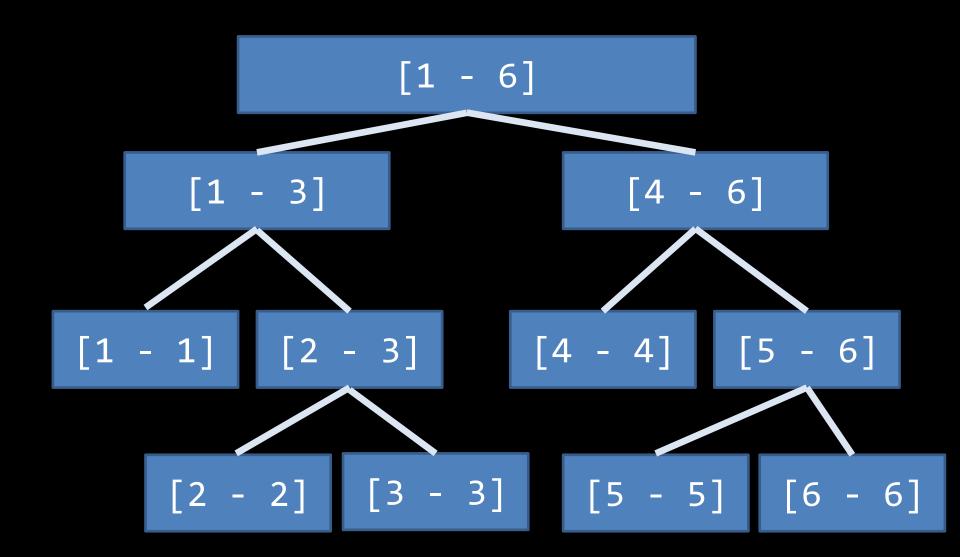
#### Overview

- Segment tree is a data structure that supports operations on 'segments'
- Usually, operations runs in O(lg N)
- Often appeared in IOI

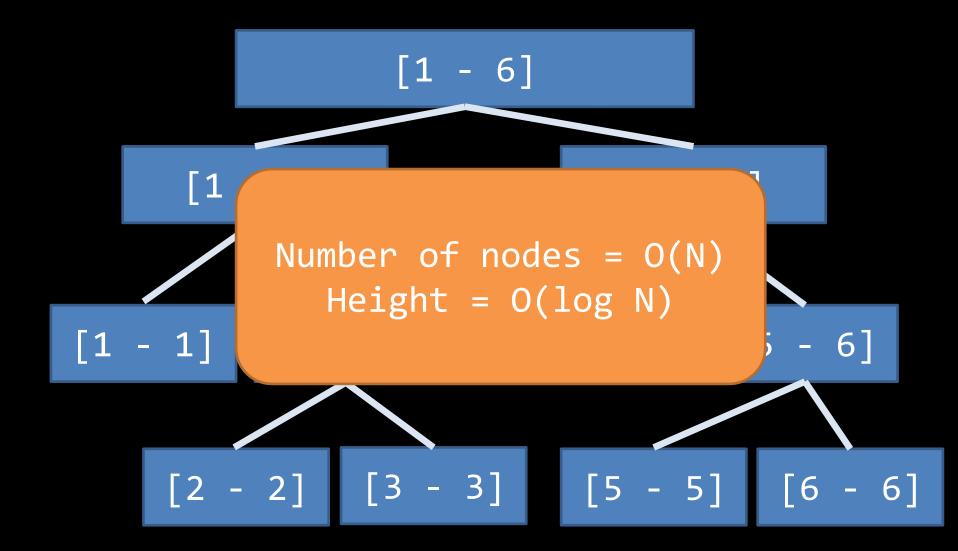
#### Structure

- Segment tree is a full binary tree
   i.e. every node has 0 or 2
   children
- The root represent the whole segment
- The segment is break into two children

#### Structure



#### Structure



- Given an array A[1..N]
- There are 2 types of operations
  - -A[x] := v
  - -What is A[l] + A[l+1] + ... + A[r]? (Sum of A[l..r])

```
Initialize
Initial
        Sum_{s} = 6
                                  Sum = 15
  Sum
              Sum_{\star} = 5
```

#### Initialize

- There are O(N) nodes
- Build the tree bottom-up
- Time complexity: O(N)

#### Query

1 = 4

r = 6

#### Query

$$[1 - 6]$$
  
Sum = 21

$$\begin{bmatrix} 1 - 3 \end{bmatrix}$$

$$Sum_{s} = 6$$

$$[1 - 1]$$
  $[2 - 3]$   
Sum = 1 Sum = 5

$$[2 - 2]$$

$$[4 - 4]$$
  $[5 - 6]$   
Sum = 4 Sum = 11

Sum = 15

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 5 Sum = 6

#### Q: What is sum A[4..6]? Query Query A: I don't know. 1 = 4 Ask my children. r = 6Sum = 21 $Sum_{.} = 6$ Sum = 15Sum $Sum_{\star} = 5$ Sum Sum =

```
Query
              Q: What is sum
                                    Q: What is sum
1 = 4
                 A[4..6]?
                                       A[4..6]?
               A: None of my
r = 6
                                     A: It is 15.
                 business.
       Sum_{.} = 6
                                  Sum = 15
             Sum_{\star} = 5
                                        Sum = 11
Sum
                             Sum
```

### Query 1 = 3 r = 5

#### Query

$$\begin{bmatrix}
 1 - 6 \\
 Sum = 21
 \end{bmatrix}
 \begin{bmatrix}
 1 - 6 \\
 Sum = 6
 \end{bmatrix}
 \begin{bmatrix}
 4 - 6 \\
 Sum = 15
 \end{bmatrix}
 \begin{bmatrix}
 1 - 1 \\
 Sum = 5
 \end{bmatrix}
 \begin{bmatrix}
 4 - 4 \\
 Sum = 4
 \end{bmatrix}
 \begin{bmatrix}
 5 - 6 \\
 Sum = 11
 \end{bmatrix}$$

$$[2 - 2]$$
  $[3 - 3]$   
Sum = 2 Sum = 3

```
Q: What is sum
                                   A[3..5]?
Query
                    Query
                               A: I don't know.
1 = 3
                               Ask my children.
r = 5
                   Sum = 21
       Sum_{.} = 6
                                 Sum = 15
Sum
             Sum_{\star} = 5
                            Sum
                                        Sum =
```

#### Query Q: What is sum Q: What is sum 1 = 3A[3..5]? A[3..5]? r = 5A: I don't know. A: I don't know. Ask my children. Ask my children. Sum = 6Sum = 15 $Sum_{2} = 5$ Sum = 11Sum Sum

```
Query
                      Query
  1 = 3
  r = 5
                    Q: What is sum
Q: What is sum
   A[3..5]?
                      A[3..5]?
                   A: I don't know.
A: None of my
                   Ask my children.
  business.
               Sum_{2} = 5
                                         Sum
   Sum
```

## Query 1 = 3 r = 5

#### Query

Sum = 21

```
[1 - 3]
Sum = 6
```

[4 - 6] Sum = 15

```
Q: What is sum A[3..5]?
A: None of my
```

None of my business.

A[3..5]? A: I can tell A[3..3] is 3.

Q: What is sum

[5 - 6]Sum = 11

$$[3 - 3]$$

$$Sum = 3$$

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 5 Sum = 6

# Query

#### Query

```
1 = 3
r = 5
                        Q: What is
                                        Q: What is
                   Sum
                        sum A[3..5]?
                                       sum A[3..5]?
                                        A: I don't
                          A: I can
                       tell A[4..4]
                                       know. Ask my
       Sum = 6
                           is 4.
                                        children.
            Sum_2 = 5
                                       Sum
Sum
```

#### Query Query 1 = 3r = 5Sum = 21Q: What $Sum_{.} = 6$ Q: What is is sum sum A[3..5]? A[3..5]? A: I can A: None tell A[5..5] of my is 5. $Sum_2 = 5$ Sum business.

#### Query

- Summary
  - If the node lies in the query interval, add it to answer
  - If the node intersect with the query interval, go down
- We visit at most 4 nodes at each level (exclude those 'none of my business')
- Height of tree = O(log N)
- Time complexity = O(log N)

```
Update
Update
                    Sum = 21
        Sum_{s} = 6
                                  Sum = 15
             Sum_{\star} = 5
 Sum
                             Sum
                                        Sum = 11
```

```
Update
Update
                    Sum
        Sum_{,} = 6
                                  Sum = 15
              Sum_{\star} = 5
 Sum
                             Sum
                                        Sum = 11
```

```
Update
Update
                    Sum = 21
        Sum_{,} = 6
                                  Sum = 15
             Sum_{\star} = 5
 Sum
                             Sum
                                        Sum = 11
```

```
Update
Update
                     Sum
        Sum_{,} = 6
                                    Sum = 15
              Sum_{\downarrow} = 7
 Sum
                               Sum
                                           Sum = 11
```

```
Update
Update
                     Sum
        Sum_{.} = 8
                                   Sum = 15
              Sum_{\lambda} = 7
 Sum
                              Sum
                                          Sum = 11
```

```
Update
```

```
Update
                Sum = 23
      Sum_{,} = 8
                             Sum = 15
           Sum = 7
Sum
                        Sum
                                  Sum = 11
```

#### Update

- We visit 1 node at each level
- Height of tree = O(log N)
- Time complexity = O(log N)

- We draw N lines one by one on the x-axis [1, N]. Each line has integer end points.
- For each line, find the length that is visible from top

- Process the lines from newest to oldest
- For a line [l, r], we want to know the length that is covered by newer lines

- The problem reduce to: at the begin, A[1] = A[2] = ... = 0
- There are 2 type of operations
  - -A[1] = A[1+1] = ... = A[r] = 1
  - -A[1] + A[1+1] + ... + A[r] = ?

```
Initialize
Initial
                   Sum =
        Sum_{\lambda} = 0
                                Sum_=
 Sum
             Sum =
                                      Sum =
```

#### Update

l = 1

r = 4

#### Update

$$\begin{bmatrix} 1 - 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 - 3 \end{bmatrix}$$

$$Sum_{=} 3$$

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -3 \end{bmatrix}$$

Sum = 
$$0$$
 Sum =  $0$ 

$$[4 - 4]$$
  $[5 - 6]$   
Sum = 1 Sum = 0

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 0 Sum = 0

#### Update

$$l = 1$$

$$r = 4$$

#### Update

$$[1 - 6]$$

$$Sum = 4$$

$$[1 - 3]$$

$$Sum_{s} = 3$$

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -3 \end{bmatrix}$$

$$Sum = 0$$
  $Sum = 0$ 

$$[4 - 4]$$
  $[5 - 6]$   
Sum = 1 Sum = 0

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 0 Sum = 0

## Query 1 = 1 r = 5

#### Query

Sum =

Go down

$$Sum_{s} = 3$$

$$[1 - 1]$$
  $[2 - 3]$   
Sum = 0 Sum = 0

$$[2 - 2]$$
  $[3 - 3]$   
Sum = 0 Sum = 0

$$[4 - 4]$$
  $[5 - 6]$   
Sum = 1 Sum = 0

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 0 Sum = 0

#### Query Query Go down Add 3 to answer Sum\_ = 1 $Sum_{\lambda} = 3$ Sum = Sum Sum Sum = Sum

#### Query Query Sum = $Sum_{.}=1$ $Sum_{\lambda} = 3$ Add 1 to answer Sum Sum = Sum

Sum = 0 Sum = 1 Sum = 0

- 2] 
$$[3 - 3]$$
  $[5 - 5]$   $[6 - 6]$ 
= 0 Sum = 0 Sum = 0

Go down

#### Query Query Sum = $Sum_{\lambda} = 3$ $Sum_{m} = 1$ SIIM Sum Sum = Sum Add 0 answer

Sum

#### Query

1 = 2

r = 3

#### Query

We are here

$$Sum = 4$$

$$[1 - 3]$$

 $Sum_{\lambda} = 3$ 

Sum = 0 Sum = 0

$$Sum = 1$$

$$Sum_{\star} = 0$$

$$\begin{bmatrix} 2 - 2 \end{bmatrix}$$

$$Sum = 0$$

$$Sum = 0$$

$$Sum = 0$$

# Query 1 = 2 r = 3

## Query

Sum =

Go down

$$[1 - 1]$$
  $[2 - 3]$   
Sum = 0 Sum = 0

$$[4 - 4]$$
  $[5 - 6]$   
Sum = 1 Sum = 0

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 0 Sum = 0

# Query

Sum

### Query

$$\begin{bmatrix}
 1 - 6 \\
 \hline
 Go down
 \end{bmatrix}
 \begin{bmatrix}
 4 - 6 \\
 Sum = 1
 \end{bmatrix}
 \begin{bmatrix}
 2 - 3 \\
 \hline
 0 Sum = 0
 \end{bmatrix}
 \begin{bmatrix}
 4 - 4 \\
 Sum = 1
 \end{bmatrix}
 \begin{bmatrix}
 5 - 6 \\
 Sum = 1
 \end{bmatrix}$$

$$[2 - 2]$$
  $[3 - 3]$   
Sum = 0 Sum = 0

#### Query Query Sum = Sum $Sum_{m} = 1$ Add 0 to answer Sum Sum = Sum Sum =

```
Query
Query
               Sum = 4
               FAIL
         Sum
                              Sum =
Sum
```

```
Query
```

#### 1 = 2

r = 3

#### Query

Problem: the node under here is not correct

$$\begin{bmatrix} 1 - 3 \end{bmatrix}$$
Sum = 3

$$[1 - 1]$$
  $[2 - 3]$   
Sum = 0 Sum = 0

$$[2 - 2]$$
  $[3 - 3]$   
Sum = 0 Sum = 0

$$[4 - 4]$$
  $[5 - 6]$   
Sum = 1 Sum = 0

$$[5 - 5]$$
  $[6 - 6]$   
Sum = 0 Sum = 0

```
Query
1 = 2
```

#### Query

```
r = 3
```

Solution: Push the value down

$$\begin{bmatrix}
 1 - 3 \\
 Sum = 3
 \end{bmatrix}
 \begin{bmatrix}
 4 - 6 \\
 Sum = 1
 \end{bmatrix}
 \begin{bmatrix}
 1 - 1 \\
 2 - 3 \\
 Sum = 0
 \end{bmatrix}
 \begin{bmatrix}
 4 - 4 \\
 5 - 6 \\
 Sum = 0
 \end{bmatrix}
 \begin{bmatrix}
 5 - 6 \\
 5 - 6 \\
 Sum = 0
 \end{bmatrix}
 \begin{bmatrix}
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 - 6 \\
 5 -$$

#### Query Query r = 36 Push & Go down $Sum_{\lambda} = 3$ $Sum_{m} = 1$ Sum $Sum_{\star} = 2$ Sum Sum =

#### Query Query Sum = $Sum_= 1$ $Sum_{s} = 3$ Add 2 to answer $Sum_{\lambda} = 2$ Sum Sum Sum = Sum

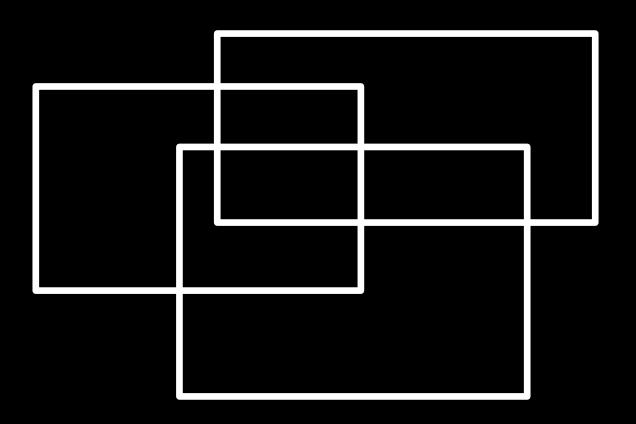
- Given A[1..n]
- There are 2 types of operations:
  - -A[x] := v
  - -Find the maximum continuous
     subsequence sum (i.e. find l, r such
     that A[l] + A[l+1] + ... + A[r] is
     maximal)

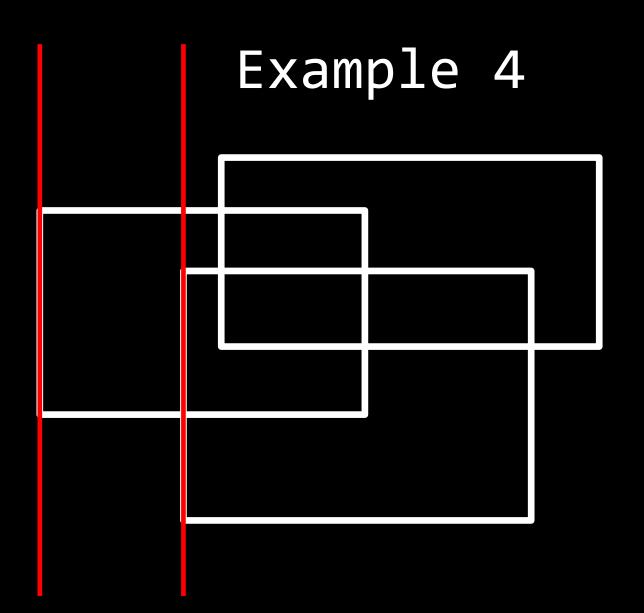
- What information a node need to store
- MCSS of the interval?
- Sum of the interval?

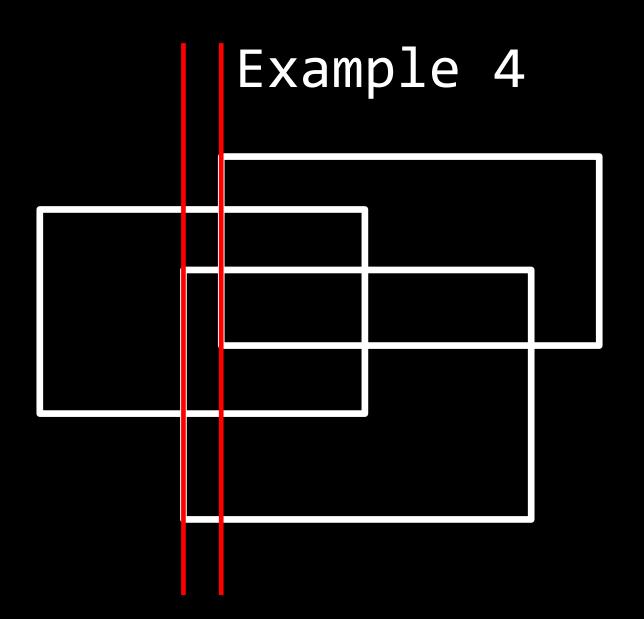
- What information a node need to store
- Sum
- Prefix MCSS
- Suffix MCSS
- MCSS

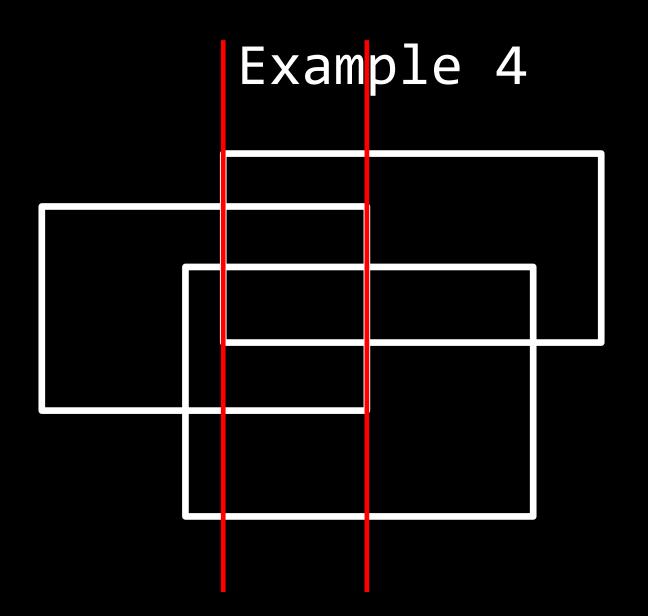
- How to update?
- Left node: trivial
- Let 1 and r be left and right children
- Sum = Sum[1] + Sum[r]
- PrefixMCSS = max(PrefixMCSS[1], Sum[1] + PrefixMCSS[r])
- SuffixMCSS = max(SuffixMCSS[r], Sum[r] + SuffixMCSS[1])
- MCSS = max(MCSS[1], MCSS[r], SuffixMCSS[1] + PrefixMCSS[r])

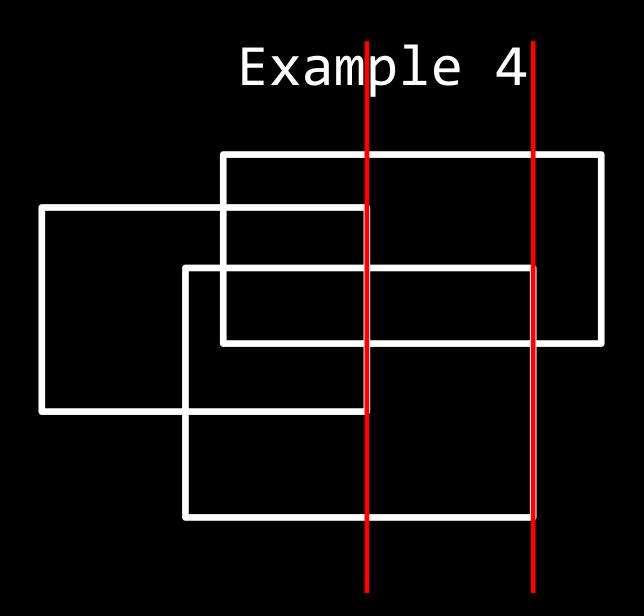
- Given N rectangles in 2D plane
- Every vertex has integer coordinates
- Every edge parallel to X or Y Axis
- Find the union area

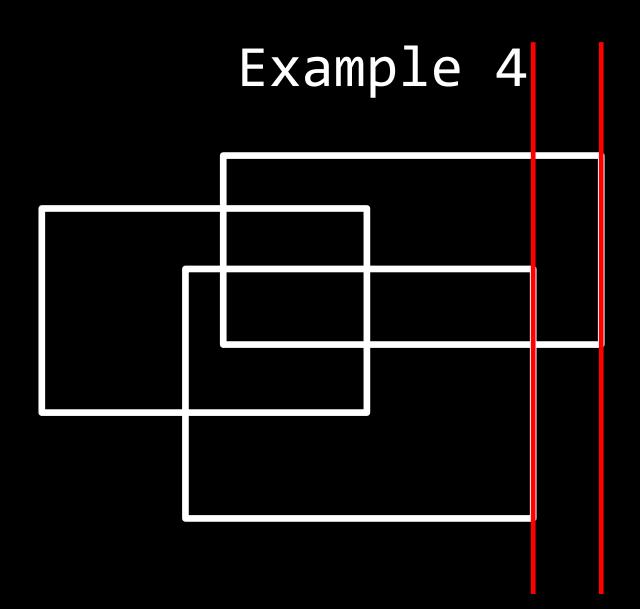












- Reduce the problem to supports the follow operations on 1..N
- Add 1 to A[1], A[1+1], ..., A[r]
- Find the numbers of non-zero entries in A[1..N]

#### Problem

- In the previous examples, the 'segment' is always a small integer
- What if the segments are large or not integer?

#### Discretization

Assume we have segments:

```
-[4.3, 7.2]
-[0.1, 10<sup>15</sup>]
-[7.2, 99]
```

• We can modify it to:

```
[2, 3][1, 5][3, 4]
```