

# Quad Tree

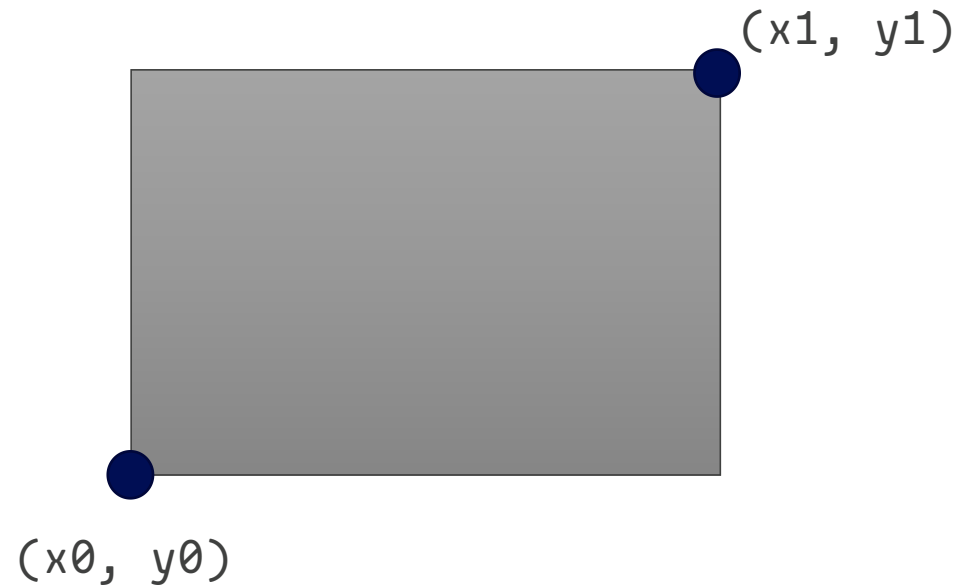
Yihao Sun

```
((λ (x y). `(,x @ ,y .edu)) 'ysun67 'syr)
```



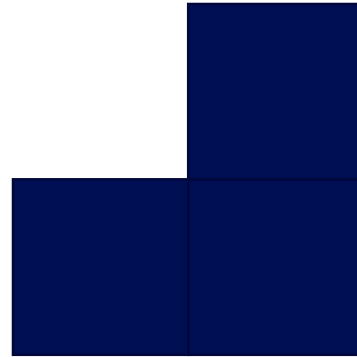
# Rectangle

use diagonal vertex coordinate to represent a `Rect`



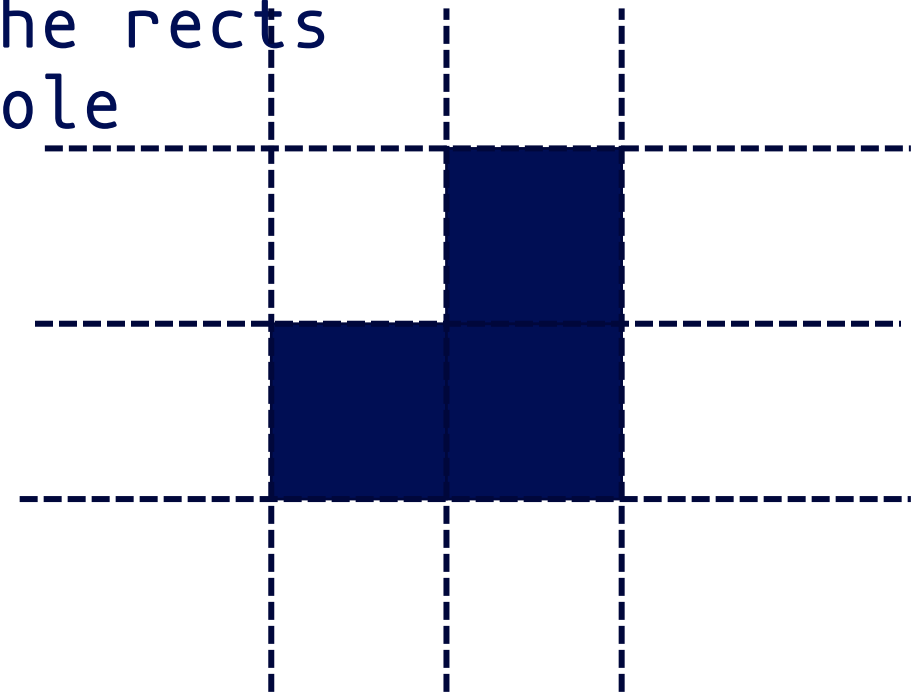
## Why We Need Quad Tree?

The following shape is a combination of Rect, but we can't just represent with vertex coordinate.



# How Quad Tree Thinks

- select a center
- Divide the whole space into 4 sub-space
- recursively repeating, until the rects formed by each center cover whole shape



# ADT of Quad Tree

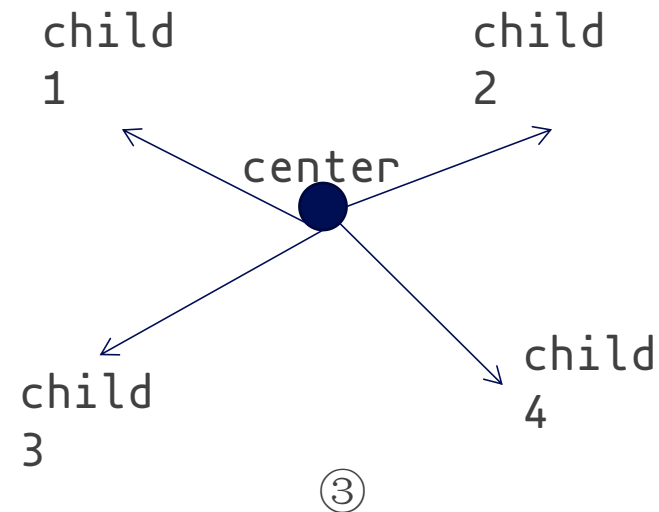
```
QuadTree := Empty           ;; empty area           ①
          | Covered         ;; covered area         ②
          | Quad (x y) QuadTree QuadTree QuadTree QuadTree ③
                                ;; quad tree has 4 children,
                                ;; (x y) is center
```



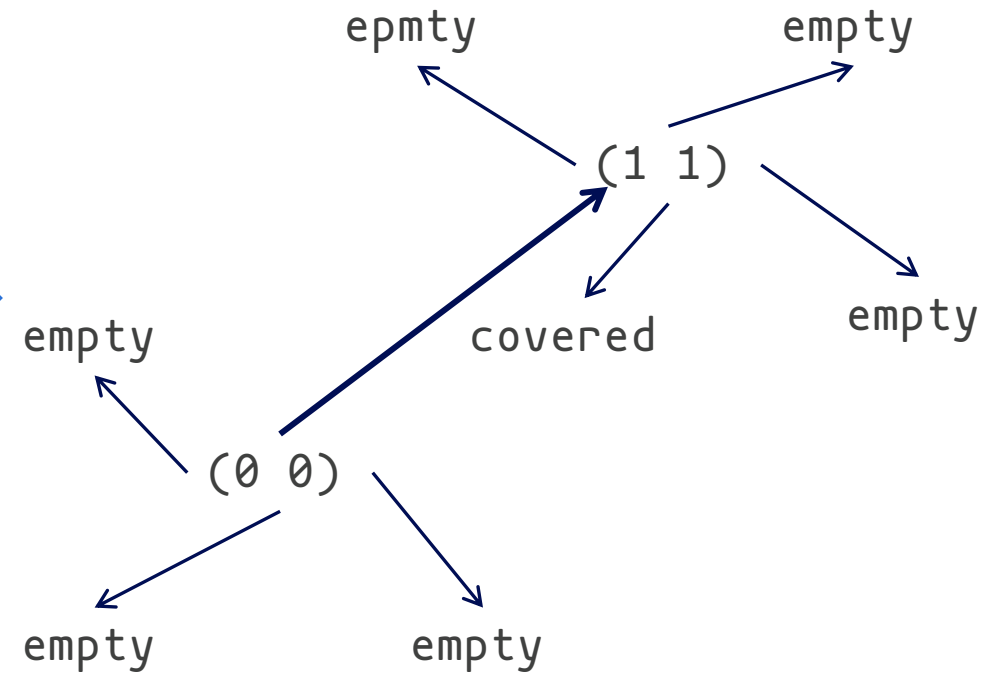
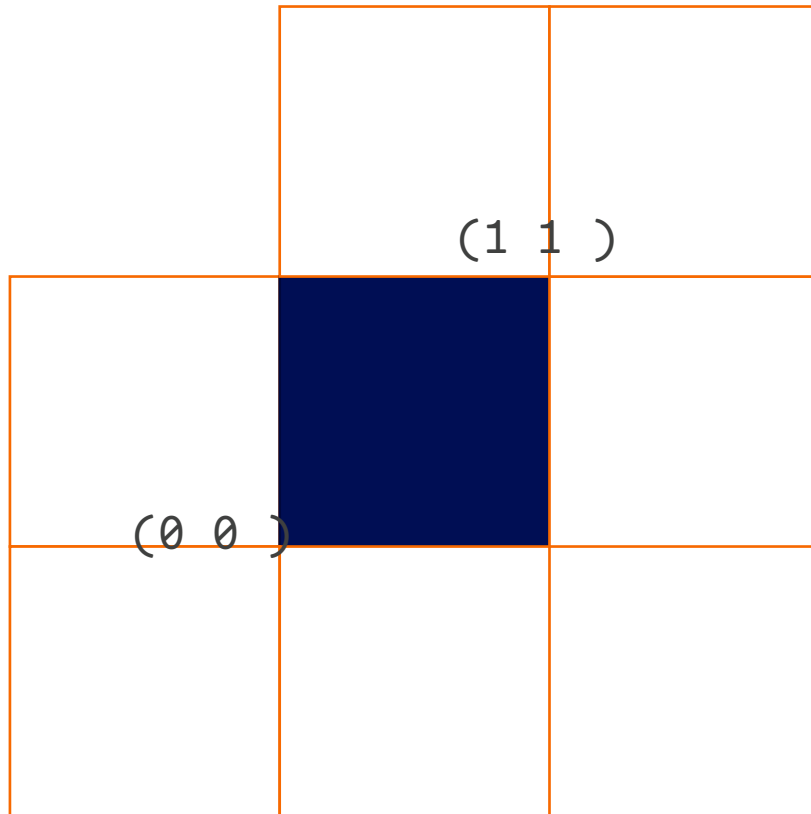
①



②

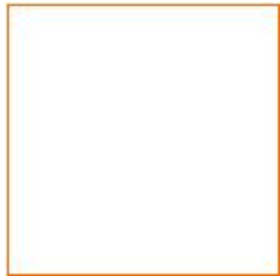


# Quad Tree Diagram

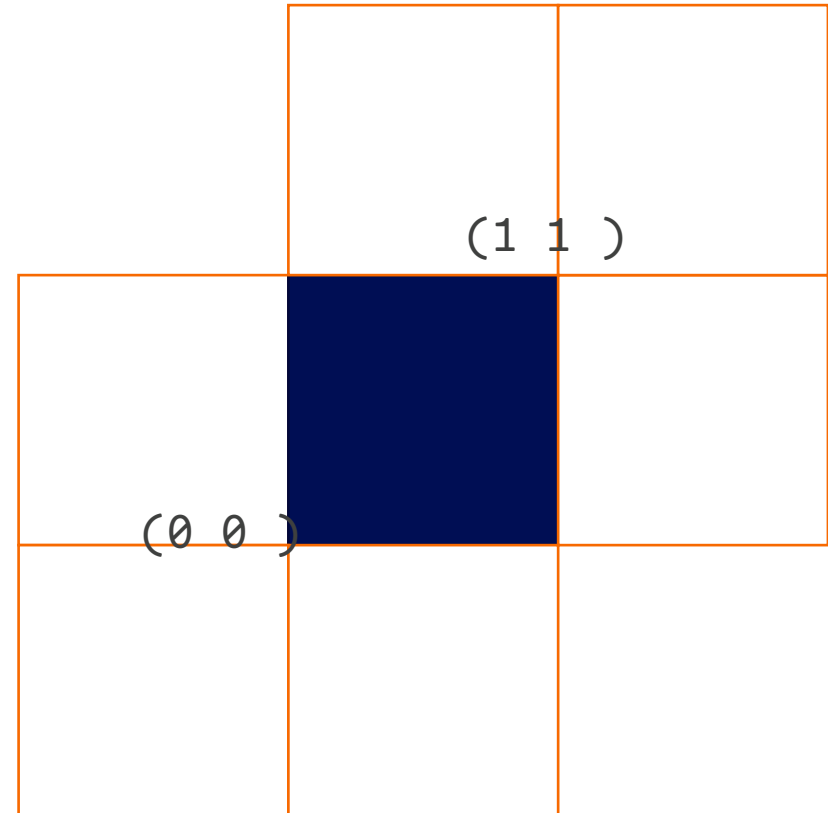


# Add Element Into Quad Tree

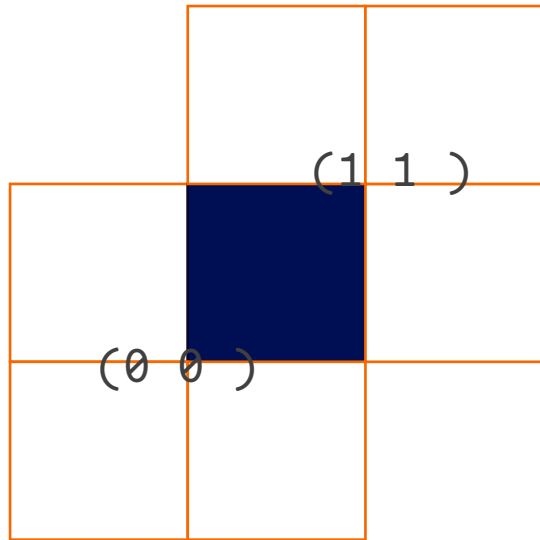
empty tree



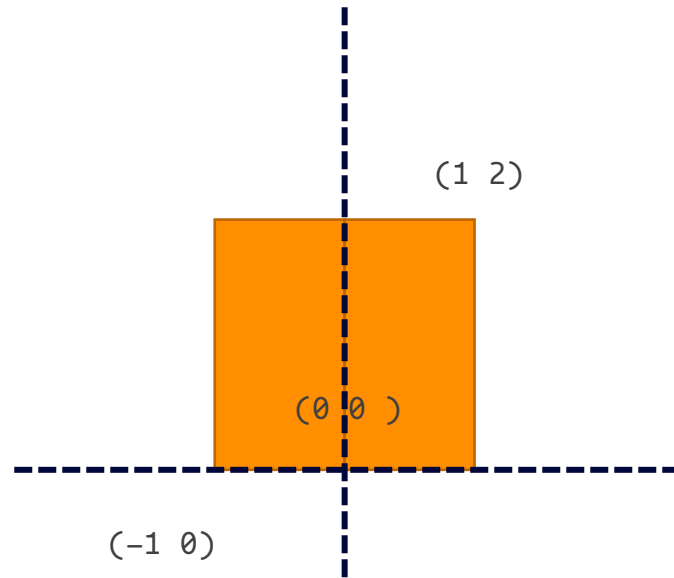
add (rect 0 0 1 1)



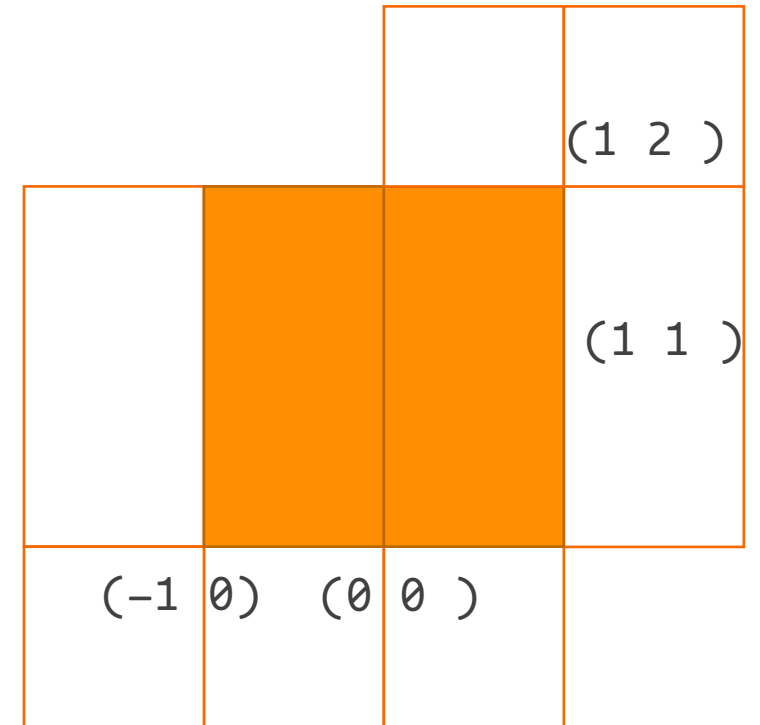
# Add Element Into Quad Tree



(rect 0 0 1 1)



add (rect -1 0 1 2)





## Function needed in code:

- `rect` : normalize a rectangle
- `rect-area` : calculate the area of a rectangle
- `quad-tree?` : predicate for quad tree
- `quad-add` : add a rect into a quad tree
- `build-quad` : build a quad tree from a list of rect
- `quad-area` : calculate all covered area in a quad tree



Thanks

