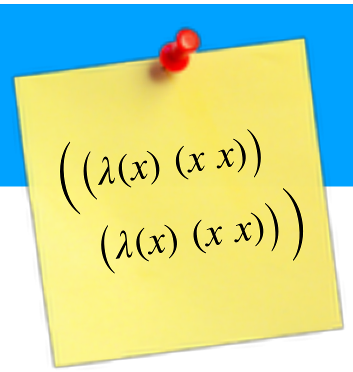
Assignment 2 Introduction

CIS 352 — Spring 2020

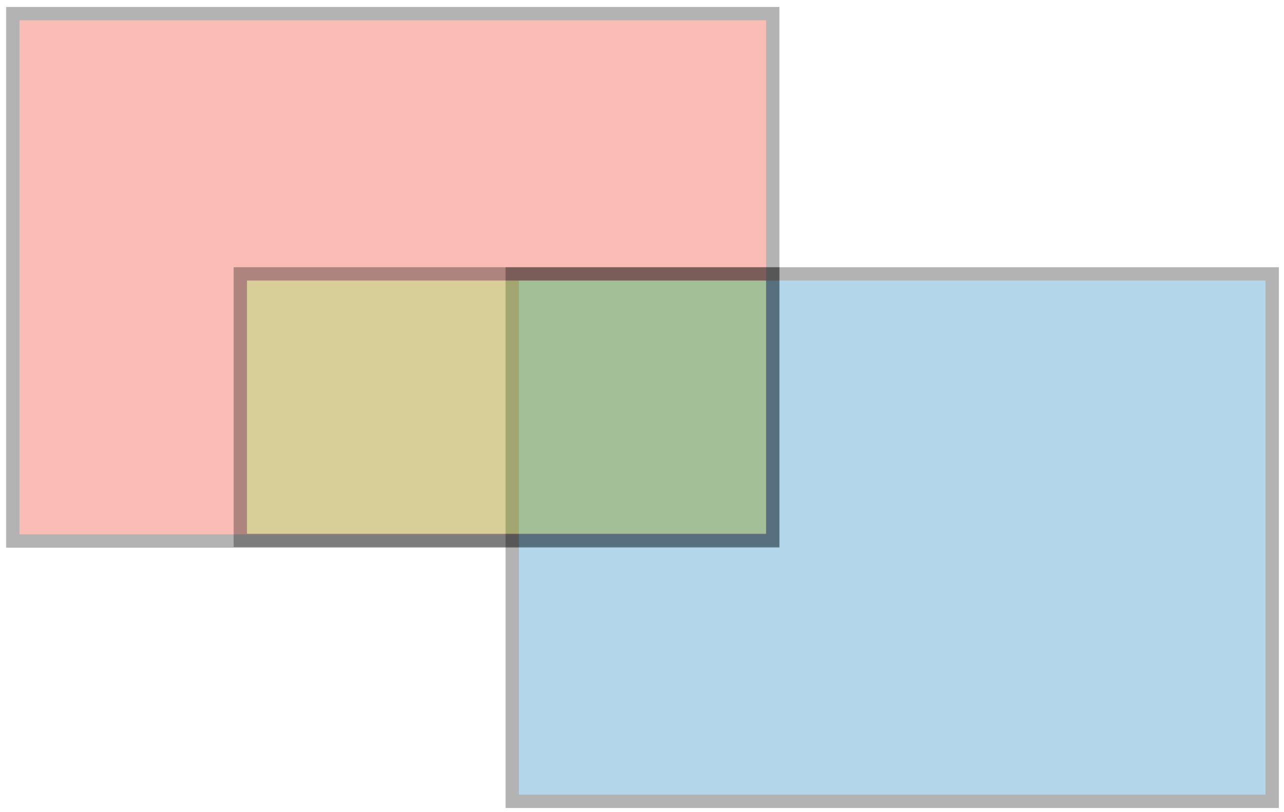
λ

Example

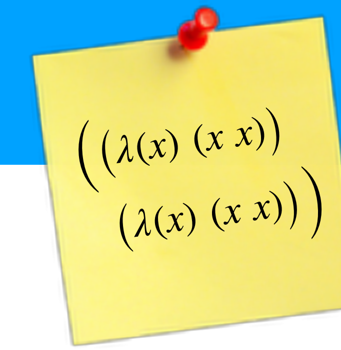


Consider that we have three rectangles, red, green, and blue

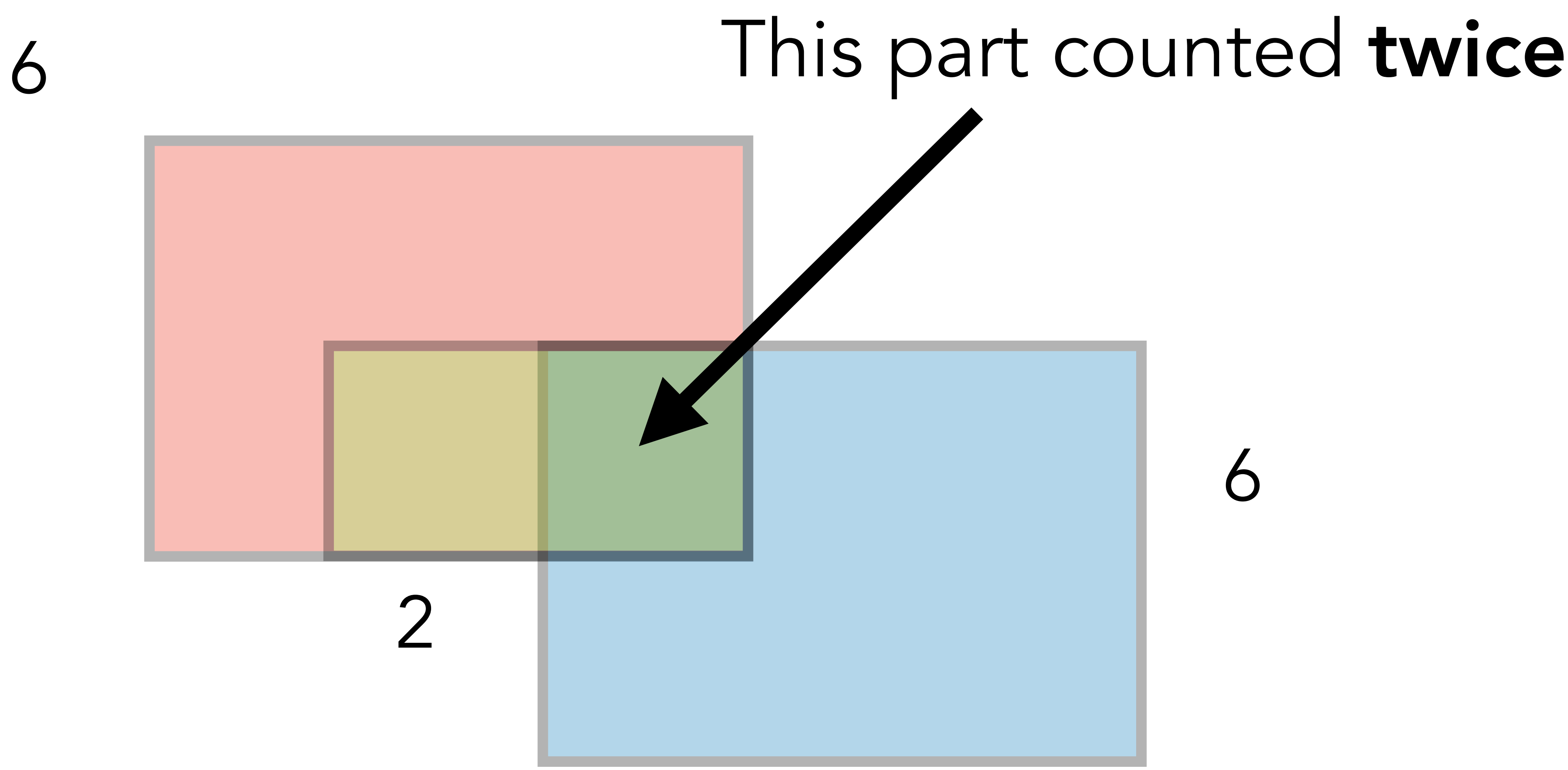
How much total area do they cover?



Example

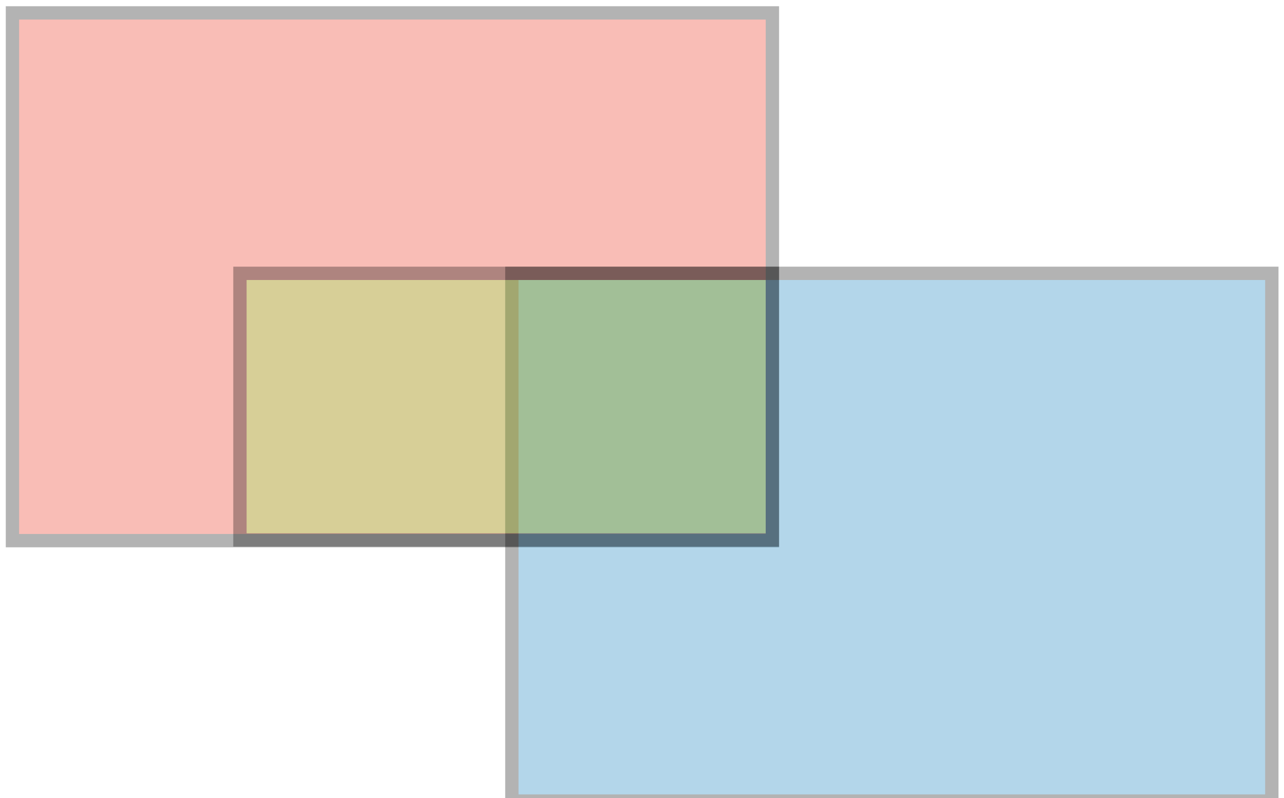


Observation: if we add areas, we get too much

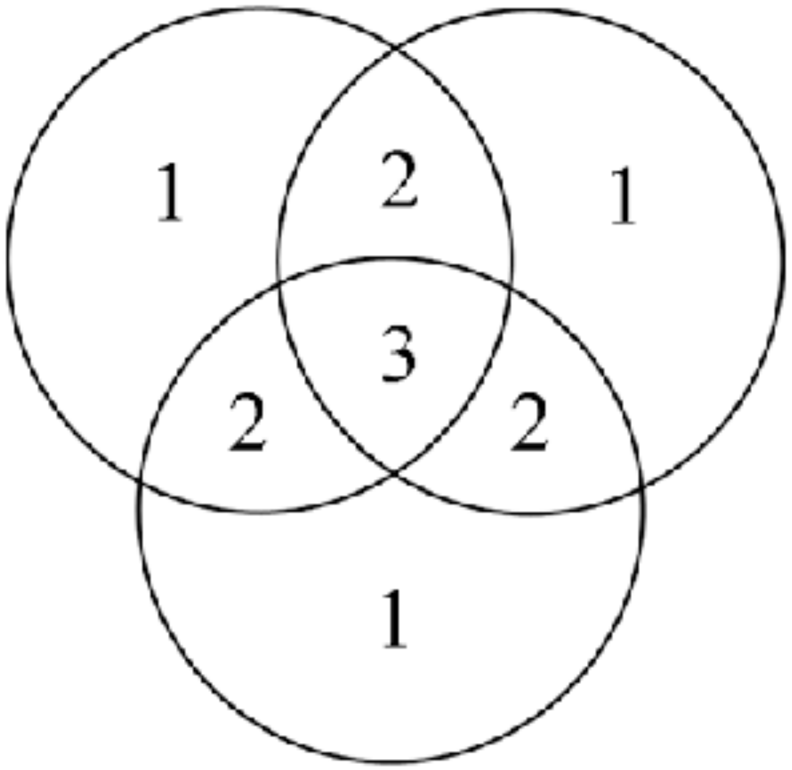


Inclusion Exclusion

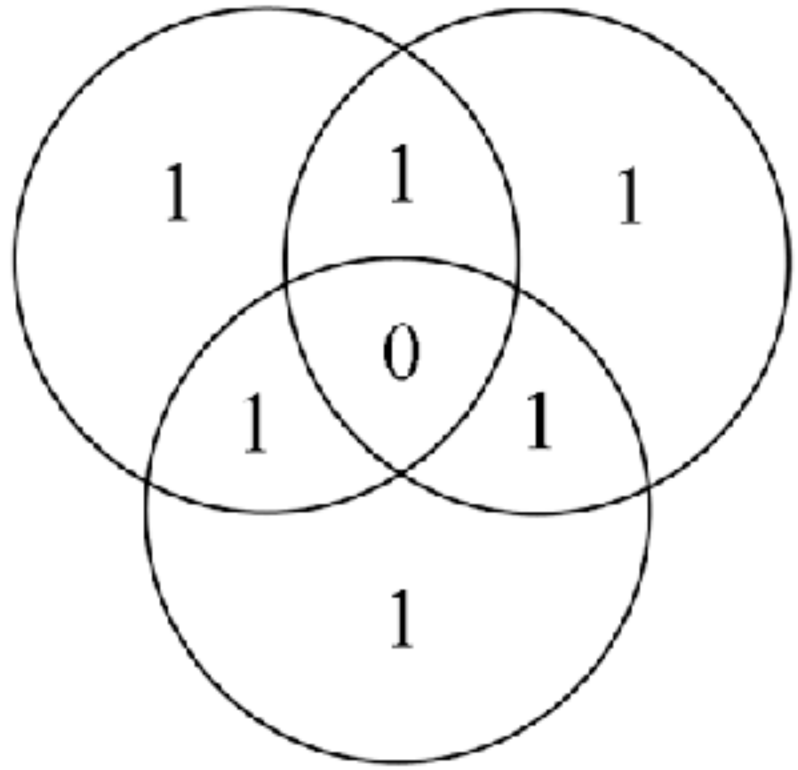
- Include (add) the sum of their areas
- Exclude (subtract) the sum of their pairwise intersection
- Include (add) the sum of their triple-wise intersections
- Exclude (subtract) the sum of quadruple-wise intersections
- Keep going until each of the N-wise intersections has been accounted for…



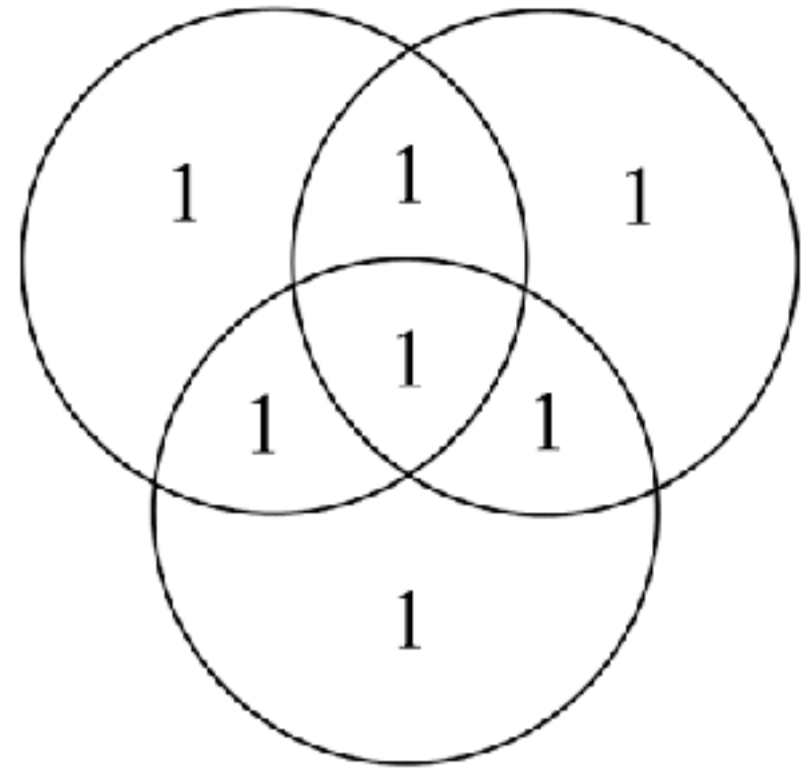
Inclusion Exclusion



$|A| + |B| + |C|$



$|A| + |B| + |C|$
$-(|A \cap B| + |A \cap C| + |B \cap C|)$



$|A| + |B| + |C|$
$-(|A \cap B| + |A \cap C| + |B \cap C|)$
$+|A \cap B \cap C|$

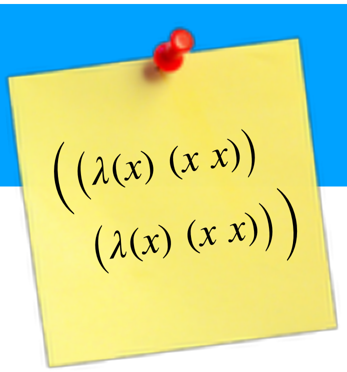
e2/e3/a2

- e2 (.25%+ bonus): rectangle operations
- e3 (.25%+ bonus): incl. excl. rectangle counting
- a2: same ideas as in e3, but can't use incl. excl.
  - Algorithm must be O(n log(n))
- Solution: use **QuadTree**

QuadTrees

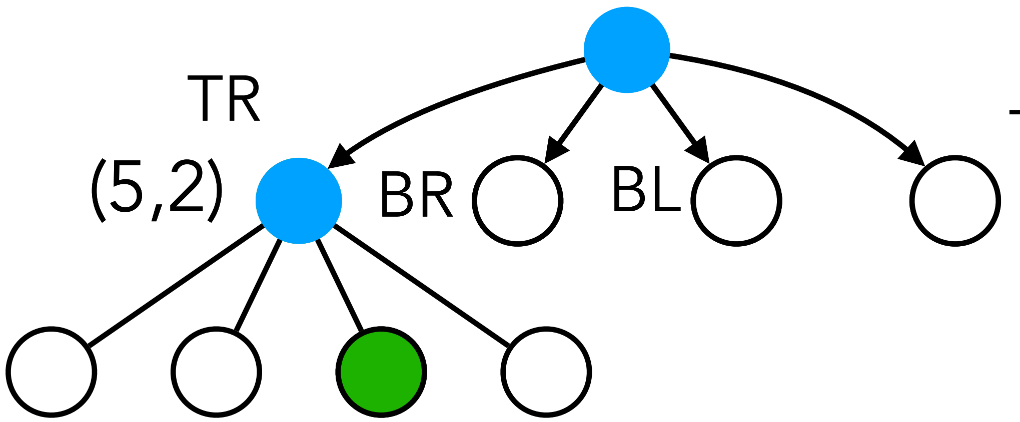
- Generalization of binary trees
  - Each node has a point and four subtrees
  - Characterizes a rectangle!
- Each node rooted at a **point** in two-dimensional space
  - Four children characterize the rectangles to the…
    - Top right, bottom right, bottom left, and top left

Example

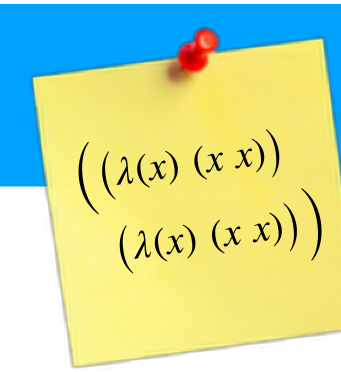


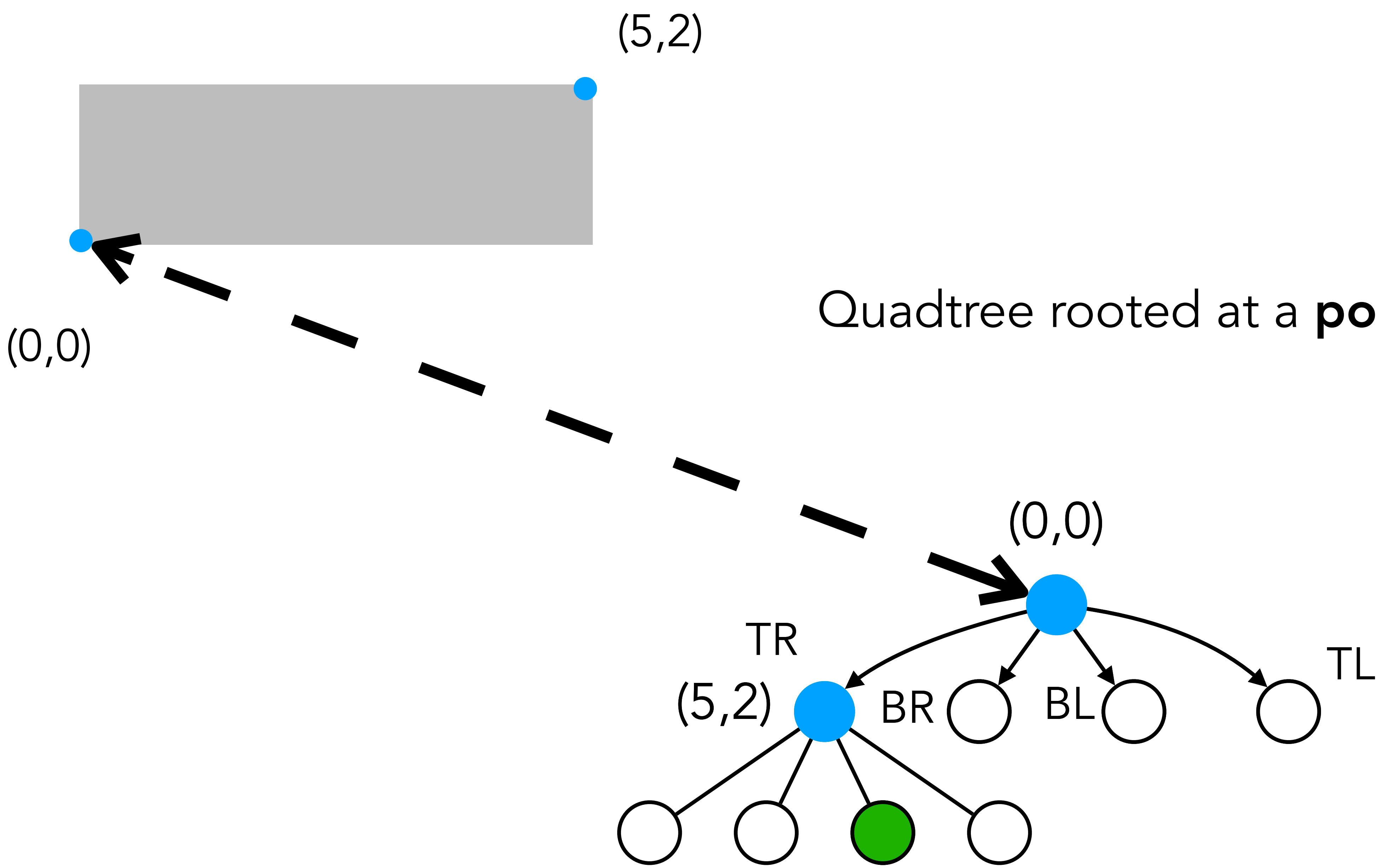
Let's say we wanted to characterize the rectangle from (0,0) to (5,2)…

Could use quadtree rooted at (0,0)



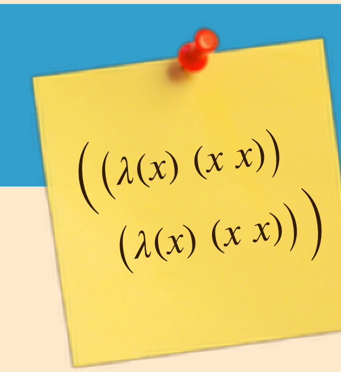
Example

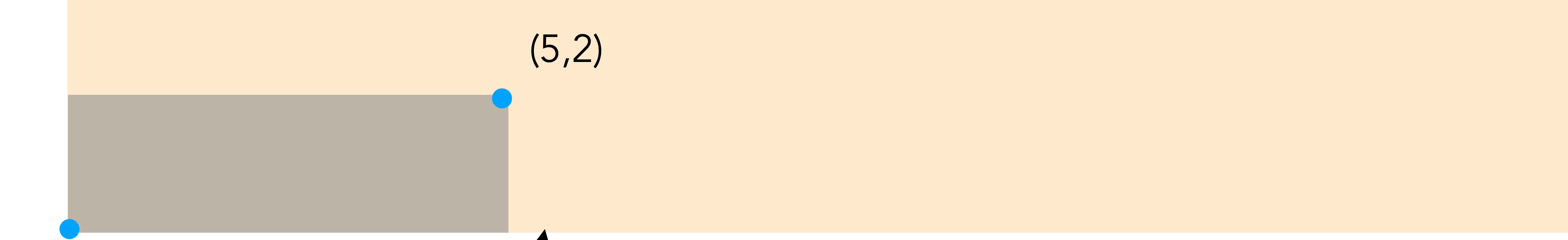




Quadtree rooted at a **point** 0,0.

Example

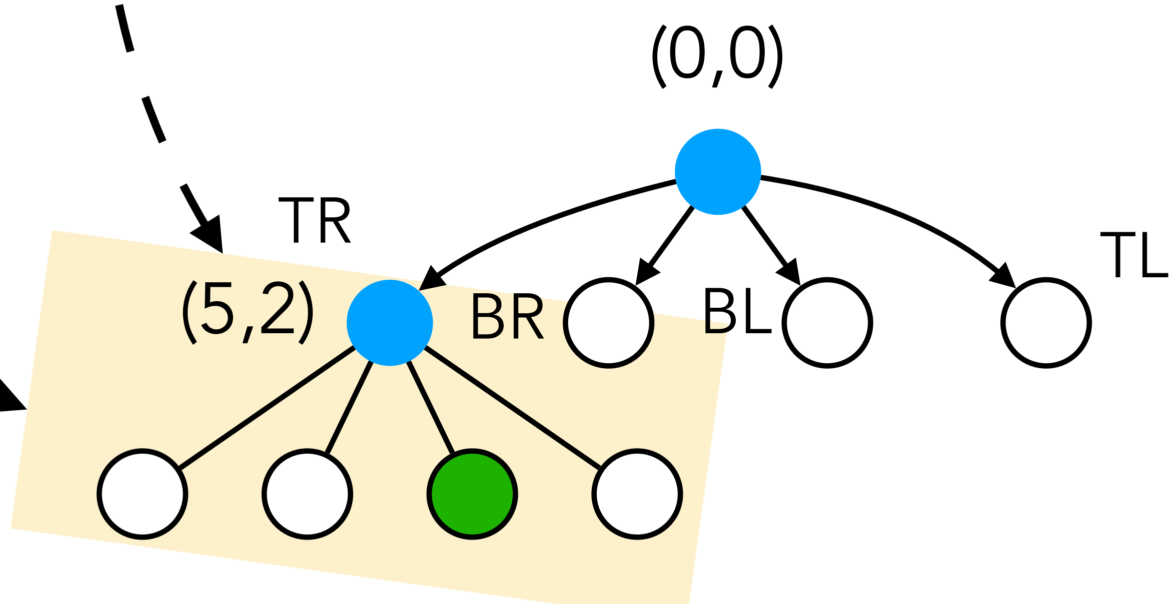




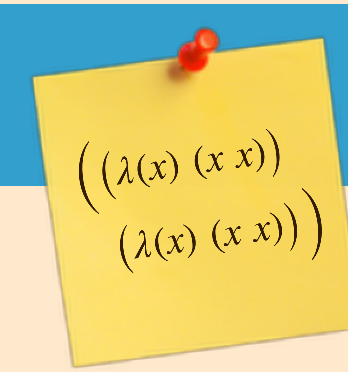
(0,0)

To the **top right** of (0,0) is this giant space…

Which is characterized by this **sub** quadtree



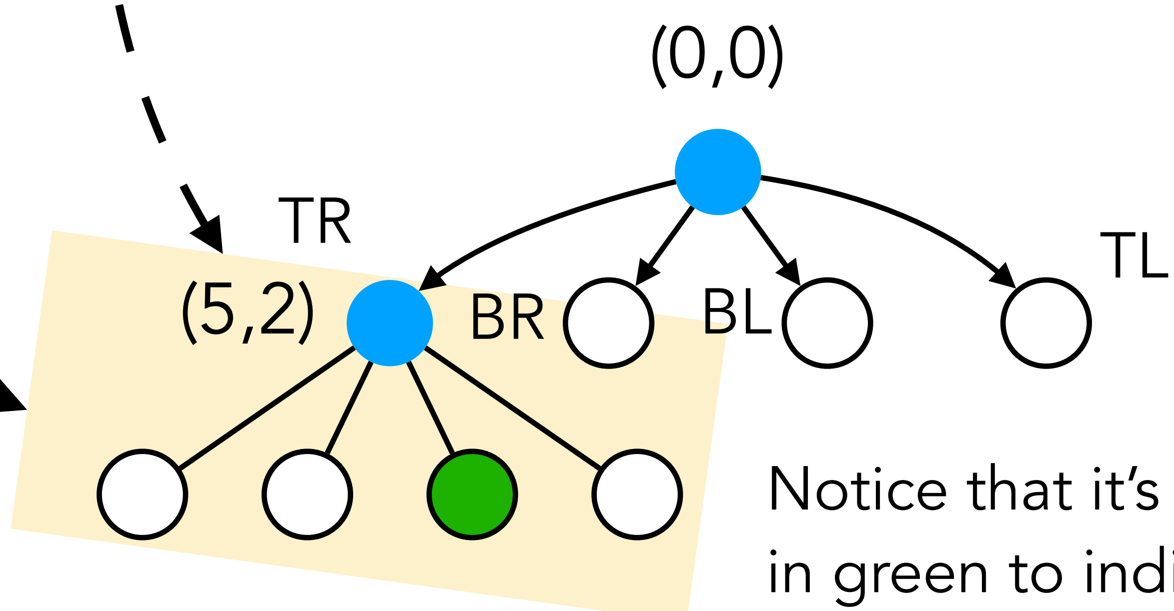
Example



To the **top right** of (0,0) is this giant space…

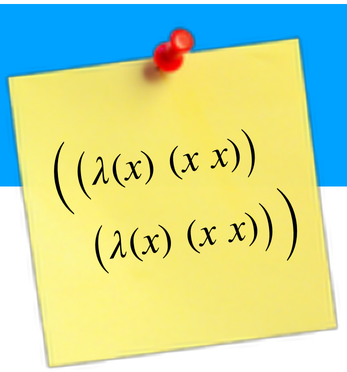
This means that the rectangle from (0,0) to (5,2) is covered

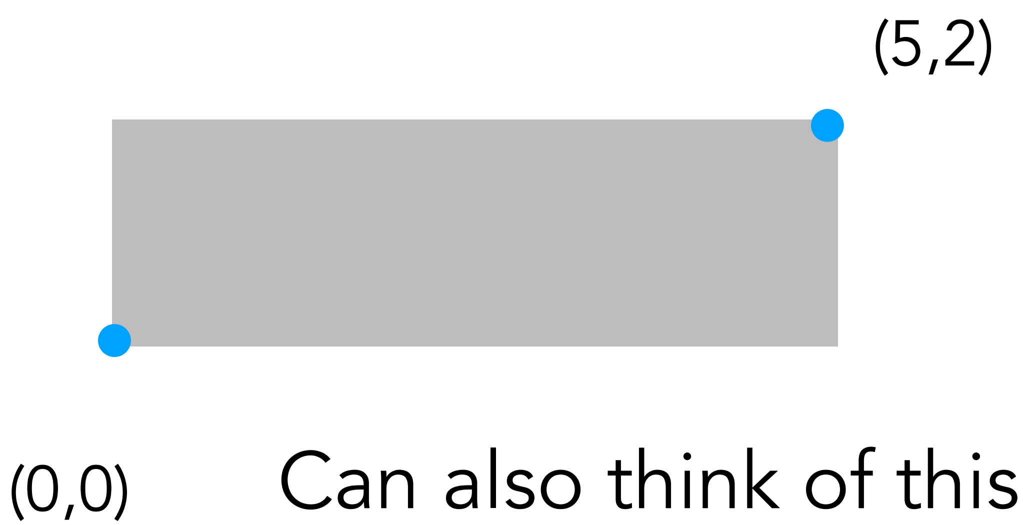
Which is characterized by this **sub** quadtree



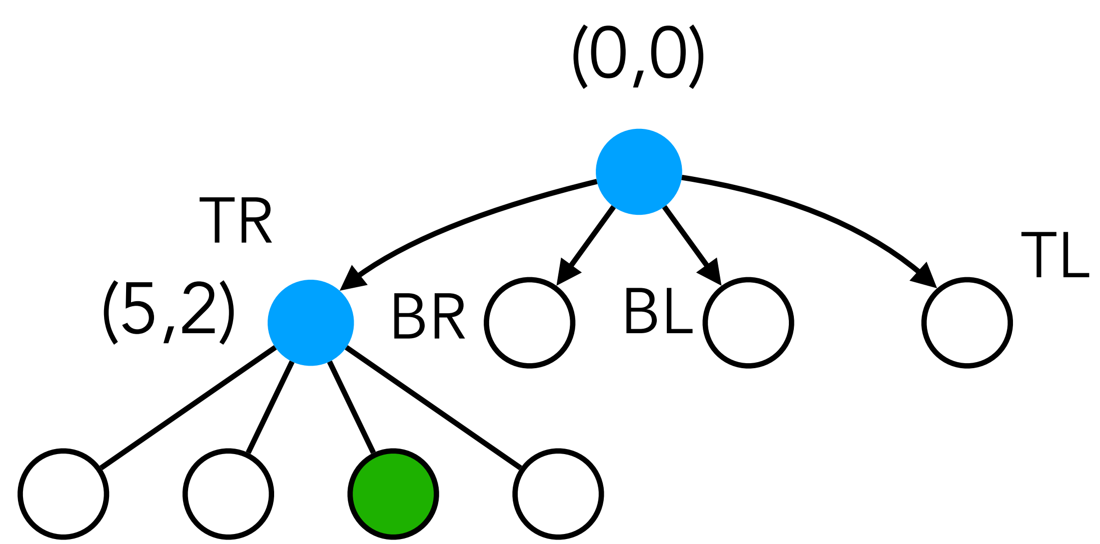
Notice that it's **bottom left** is filled in green to indicate it is **covered**

Example

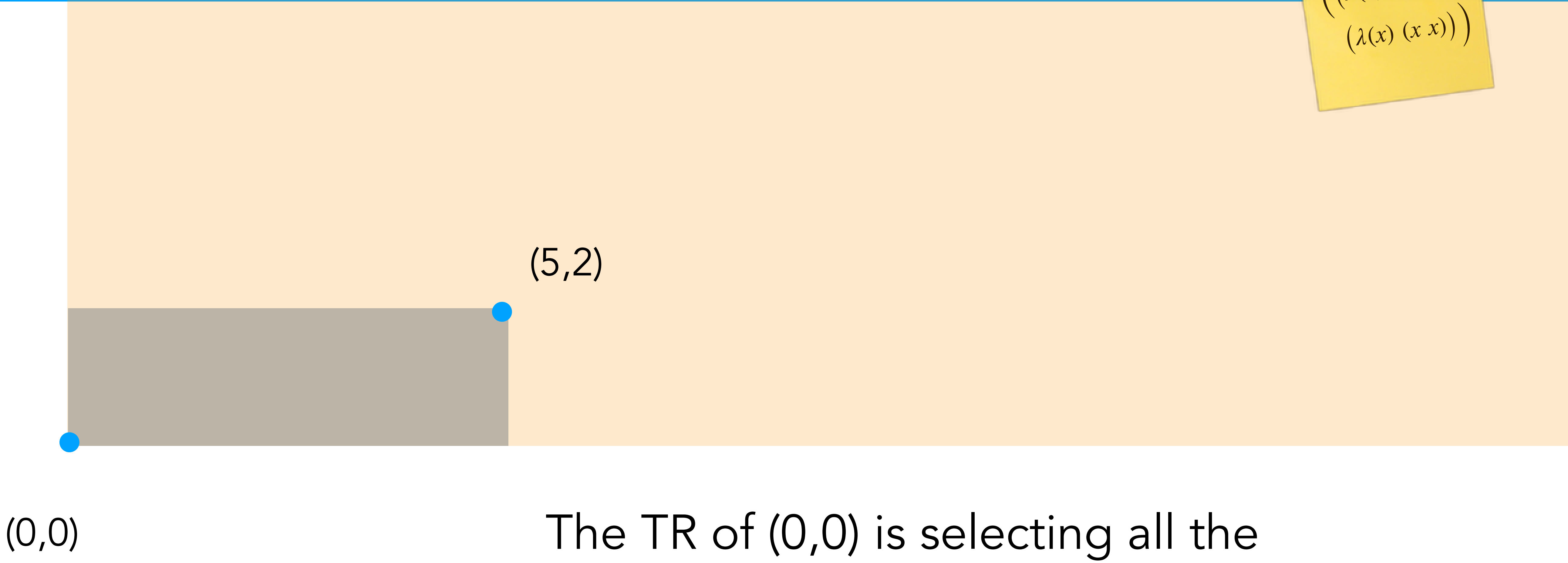




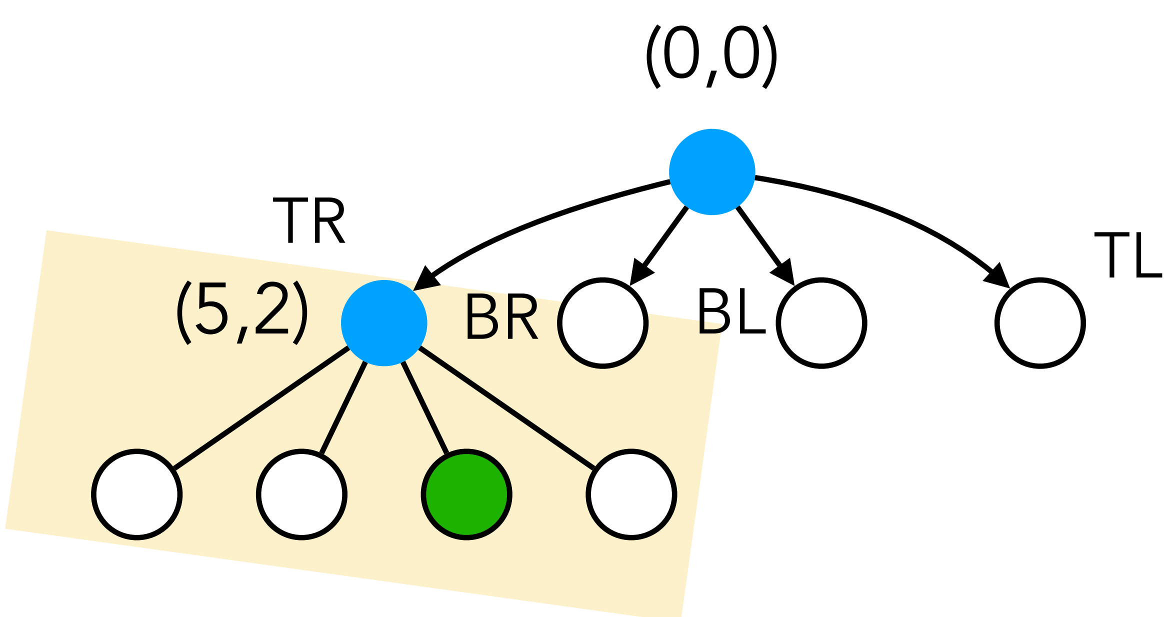
Can also think of this like a **decision diagram** on the space



Example



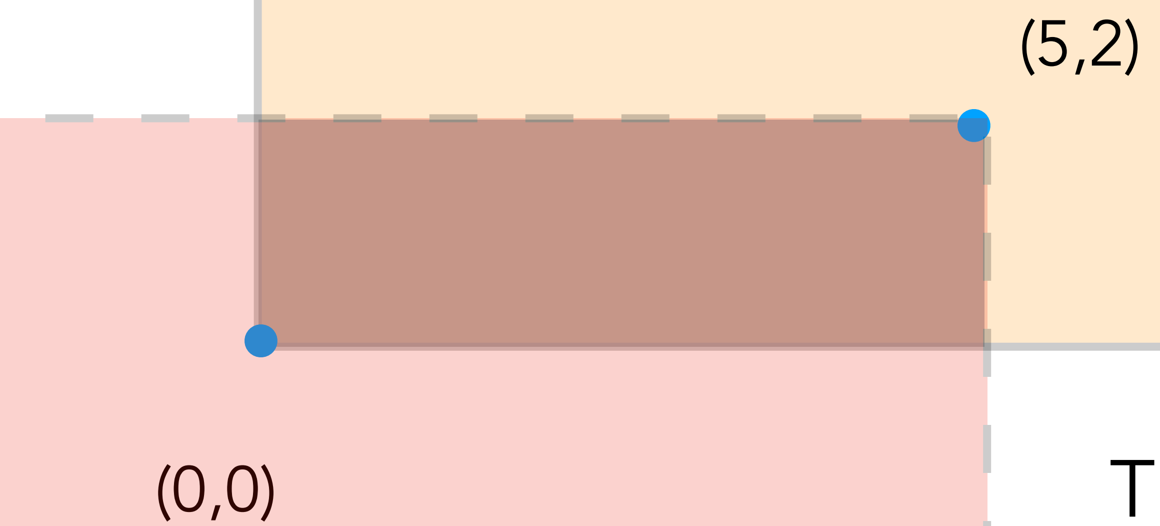
The TR of (0,0) is selecting all the space to the top right of (0,0)



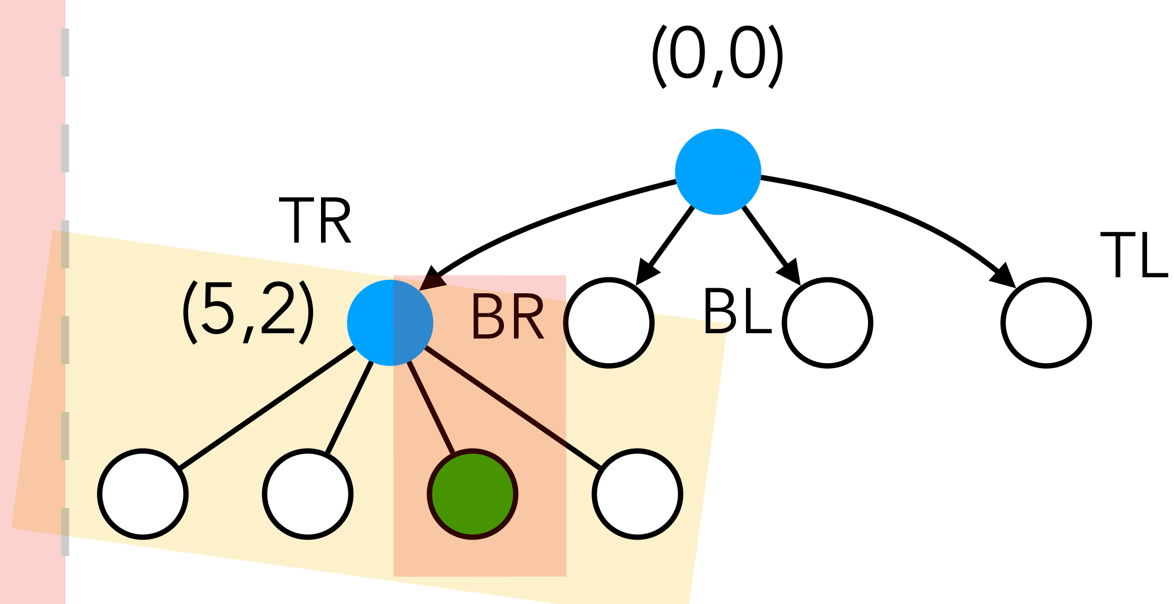
Example

$((\lambda(x)\ (x\ x))\ (\lambda(x)\ (x\ x)))$

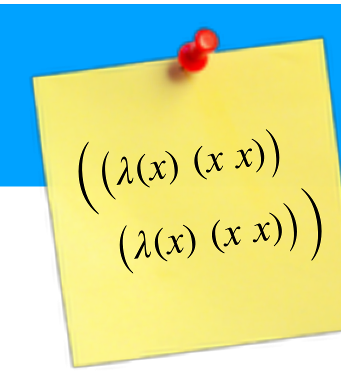
Which is already **cut off** at **this** boundary by virtue of having been nested inside the top-level quadtree



Then, the BR of (5,2) **further** limits that to the **bottom right** of (5,2)

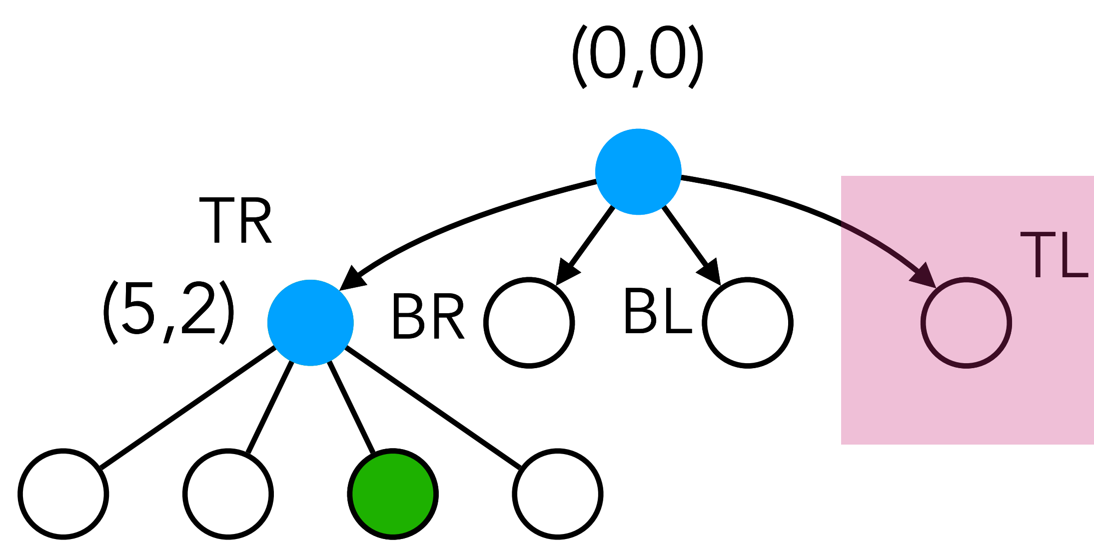


Example

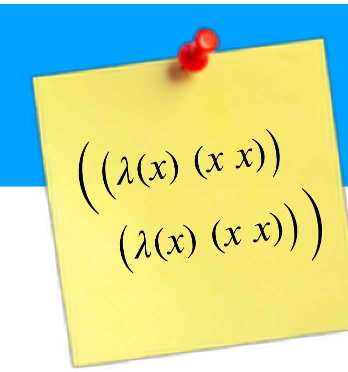


Note that the rest of the nodes are **empty**

To the **top left** of (0,0) is empty space

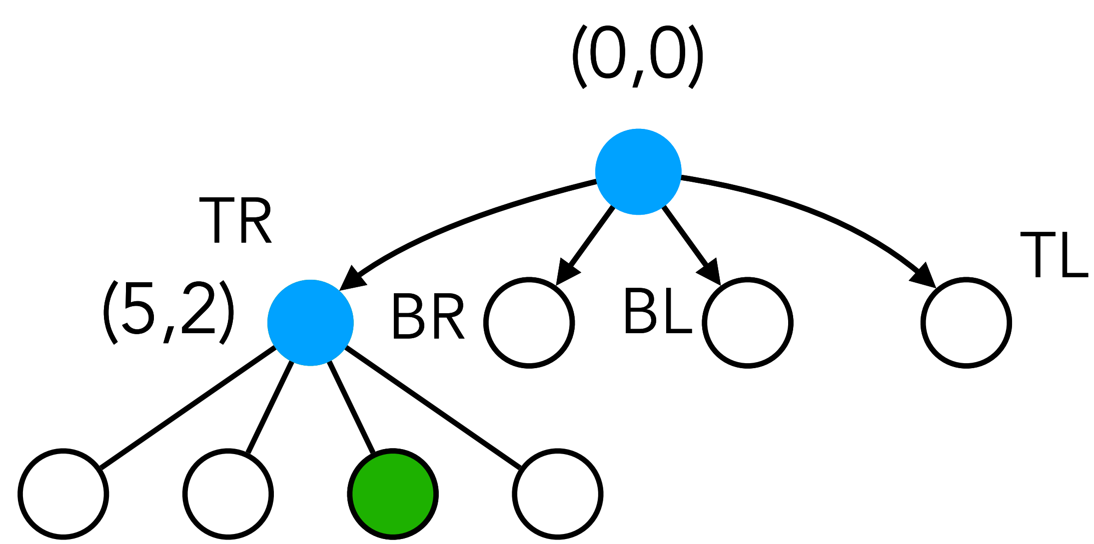


Example

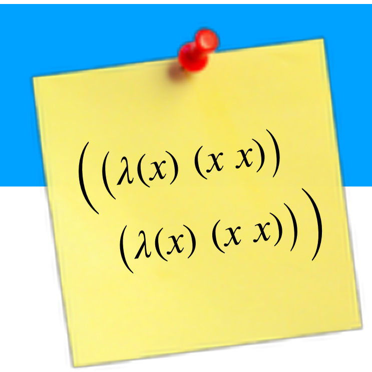


Think about how you could calculate the **area** of a quadtree

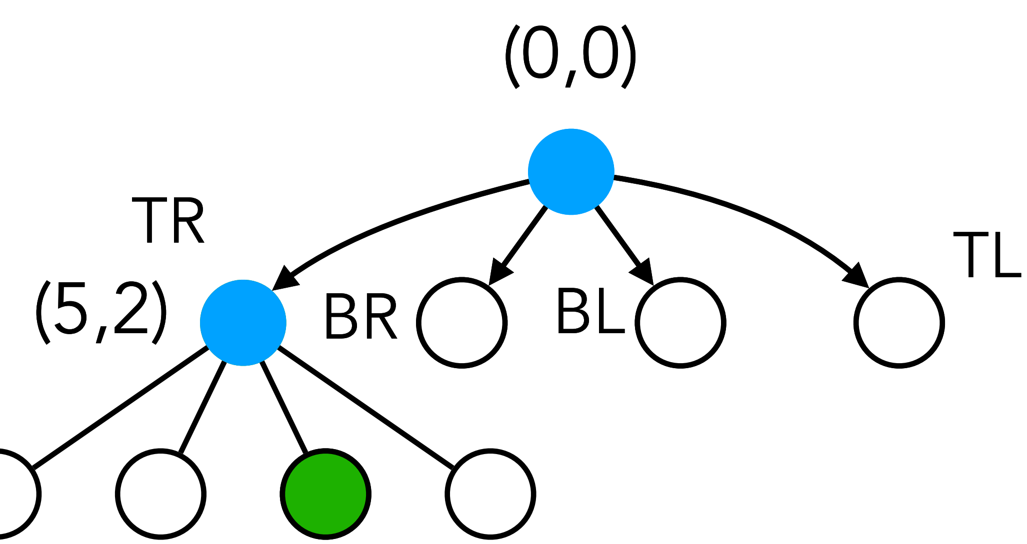
Basic intuition: always keep track of the bottom left and top right coordinate



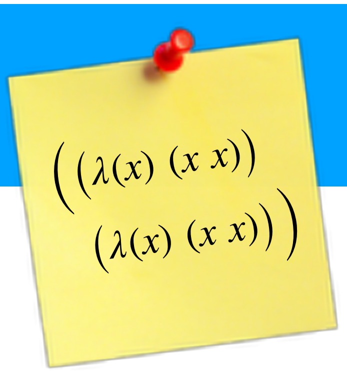
Example



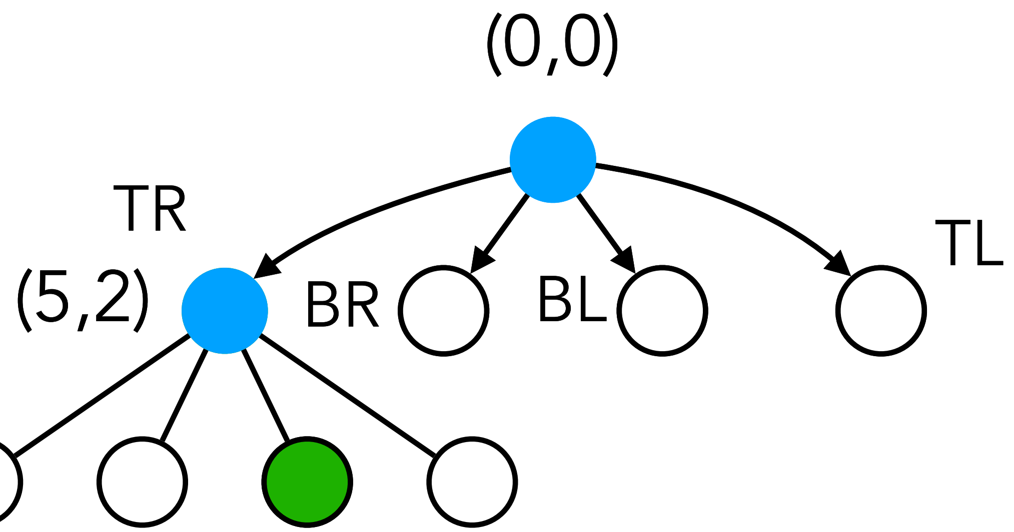
Initially, bottom left coordinate covered is (-inf, -inf), and top right is (+inf, +inf)



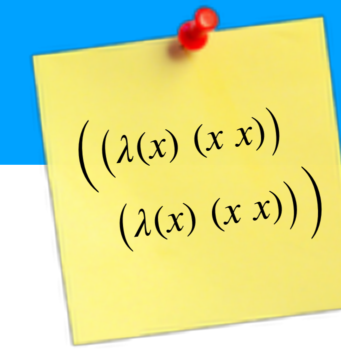
Example



Then, when we look at (0,0), we sum the areas of each of its children

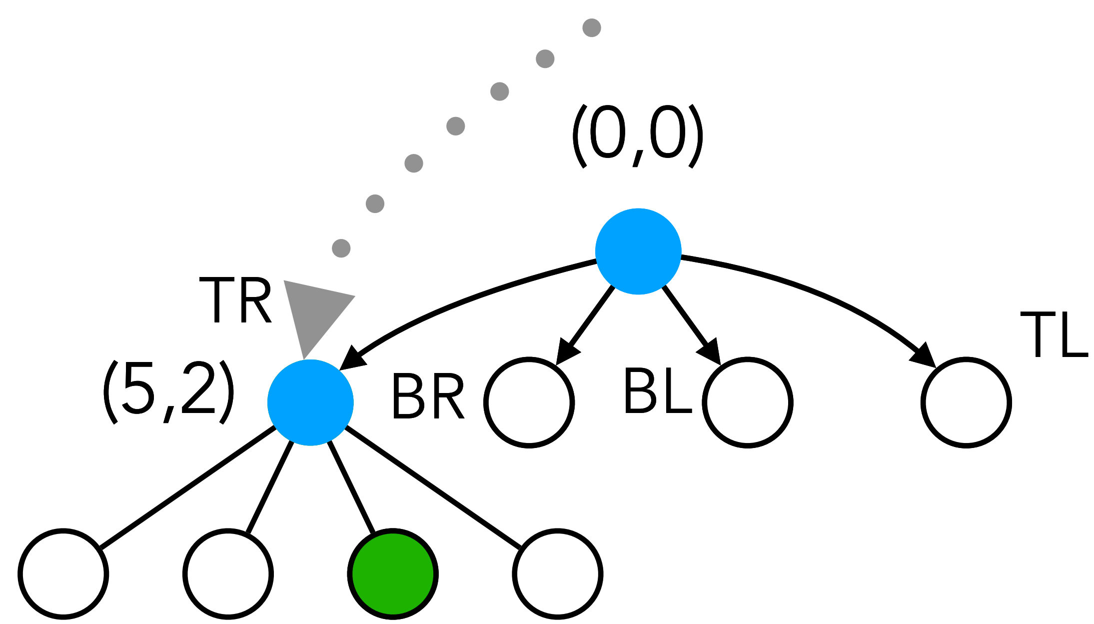


Example

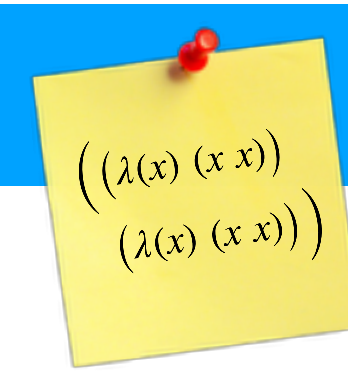


Then, when we look at (0,0), we sum the areas of each of its children

To calculate area of the top right of (0,0), we calculate it relative to the bottom left being at (0,0) rather than (-inf,-inf)

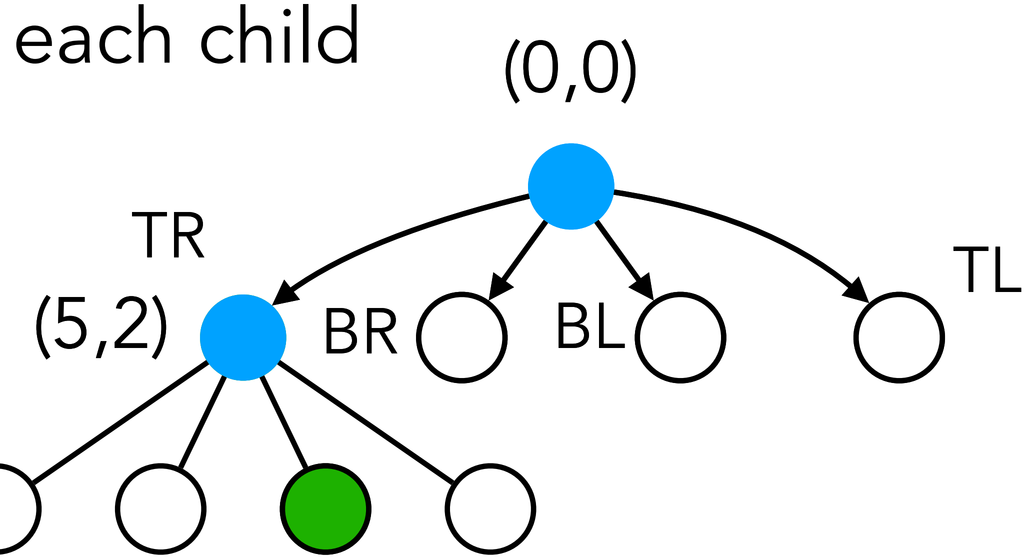


Example

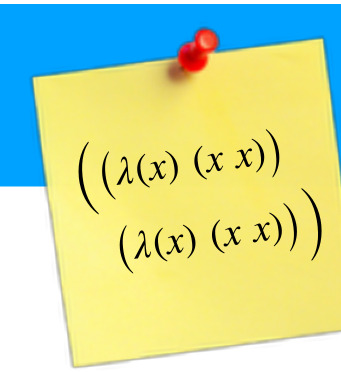


Then, when we look at (0,0), we sum the areas of each of its children

Then, when we calculate the area of (5,2), we sum the area of each child



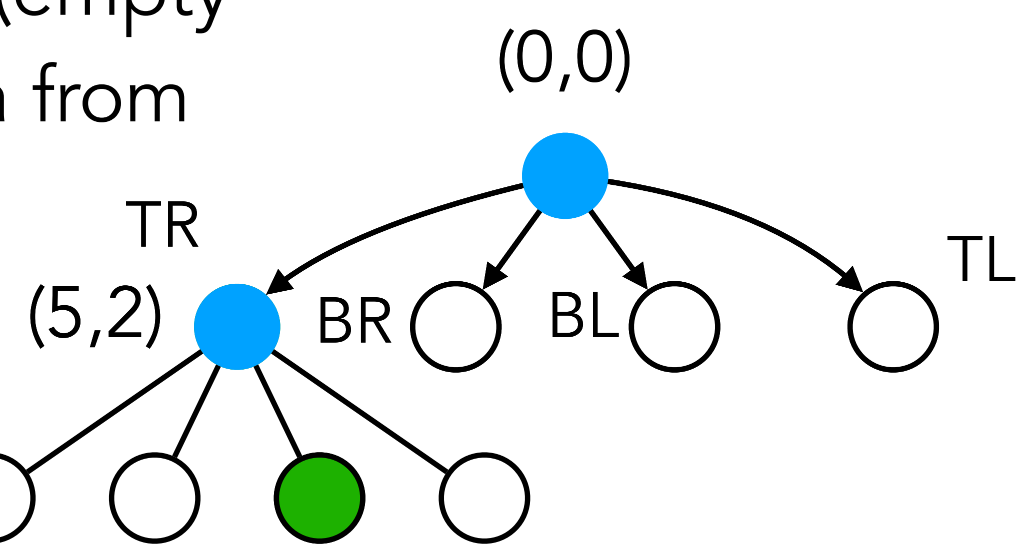
Example



Then, when we look at (0,0), we sum the areas of each of its children

Think about how you could turn this idea into Racket code…

Including the **covered** one (empty has 0 area), which has area from (0,0) to (5,2)



Solving a2

- Figure out how to **construct** quad trees
  - Three constructors/cases: empty, covered, quad node
- In these slides, we discussed how to insert into empty
- You figure out how to do others
- Figure out how to calculate the total space of a quadtree
- Define a function `(insert-rect qt r)` (or any other name) that takes a quadtree and rectangle (to insert into it) and returns a new quadtree with that rectangle inserted
- Define a function that sums the total area of a quadtree
- Insert all rectangles to build quadtree, then get total area