



# SPATIAL INDEXING

---

```
.....  
explain analyze  
select l.name  
from planet_osm_line l  
join planet_osm_roads r on  
st_intersects(l.way, r.way)  
where r.bridge = 'yes';
```

```
drop index planet_osm_line_index;
```

.....  
**Nested Loop** (cost=0.00..1442997.73 rows=14974 width=13) (actual time=0.679..1006.479 rows=3751 loops=1)

**Join Filter:** ((l.way && r.way) AND \_st\_intersects(l.way, r.way))

  Rows Removed by Join Filter: 5436393

  -> **Seq Scan** on planet\_osm\_line l (cost=0.00..1213.66 rows=29566 width=211) (actual time=0.008..2.548 rows=29566 loops=1)

  -> **Materialize** (cost=0.00..146.37 rows=184 width=309) (actual time=0.000..0.006 rows=184 loops=29566)

    -> **Seq Scan** on planet\_osm\_roads r (cost=0.00..145.45 rows=184 width=309) (actual time=0.052..0.688 rows=184 loops=1)

      Filter: (bridge = 'yes'::text)

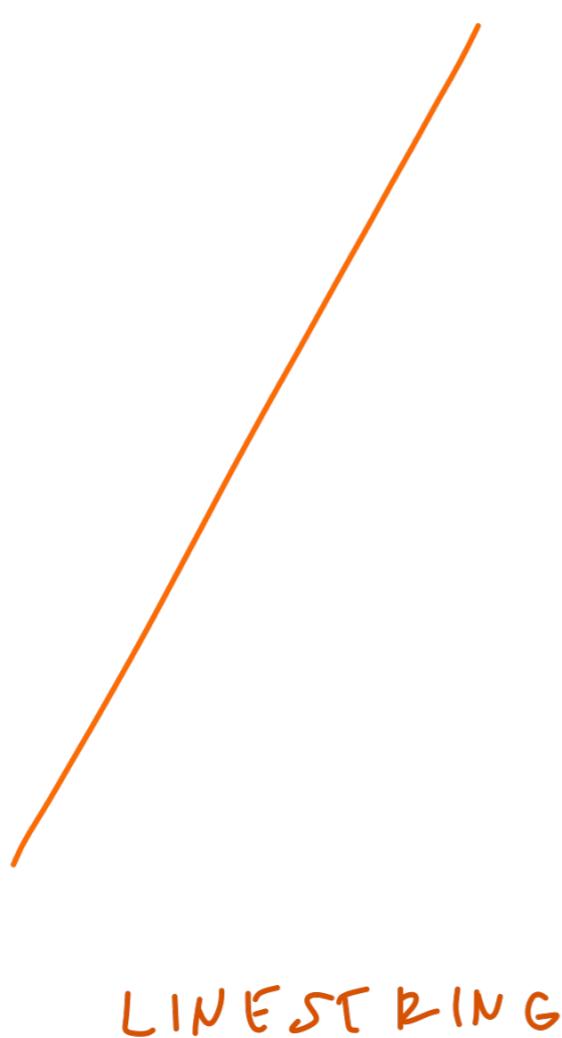
      Rows Removed by Filter: 2572

Planning time: 0.355 ms

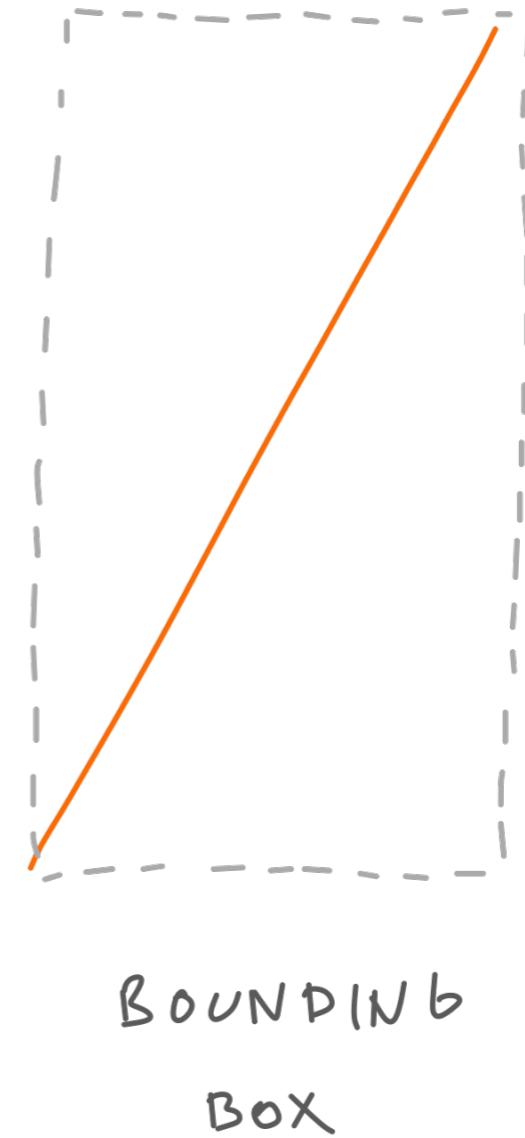
Execution time: 1006.722 ms

# && - BOUNDING BOX INTERSECTION OPERATOR

---



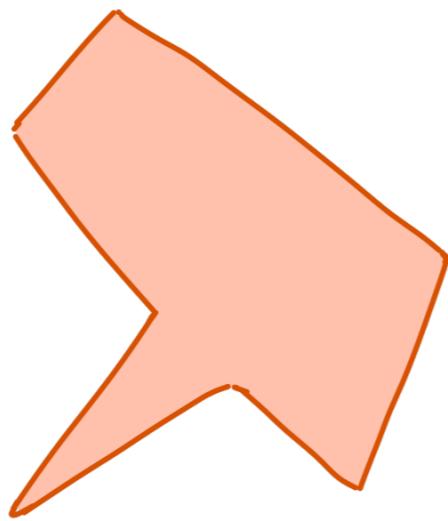
LINESTRING



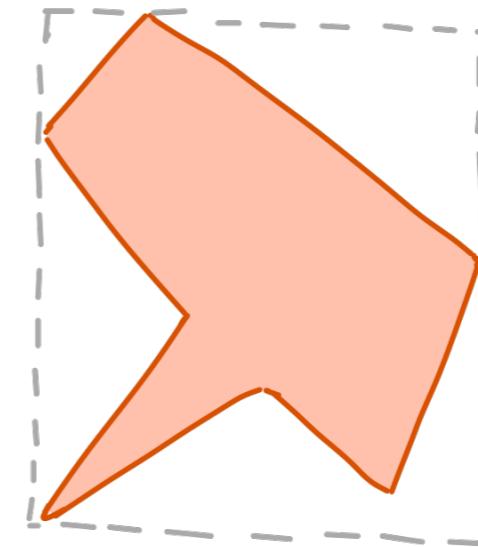
BOUNDING  
BOX

# && - BOUNDING BOX INTERSECTION OPERATOR

---



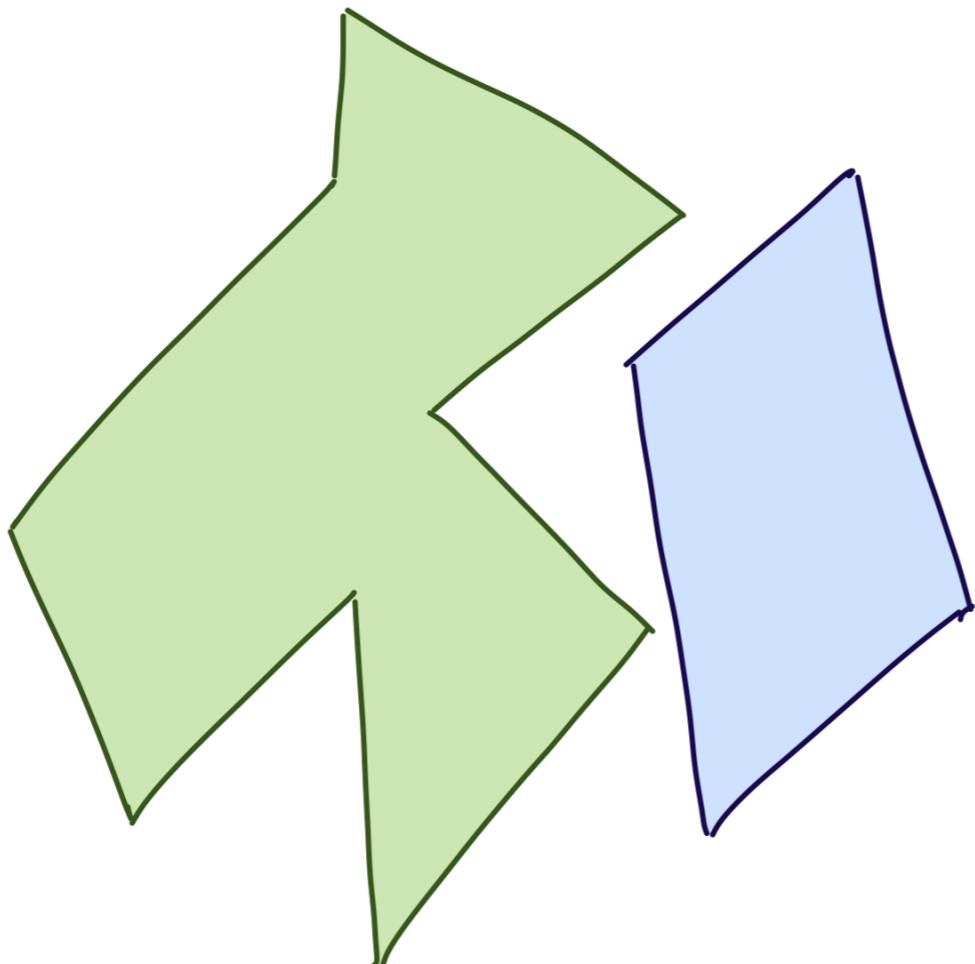
POLYGON



BOUNDING  
Box

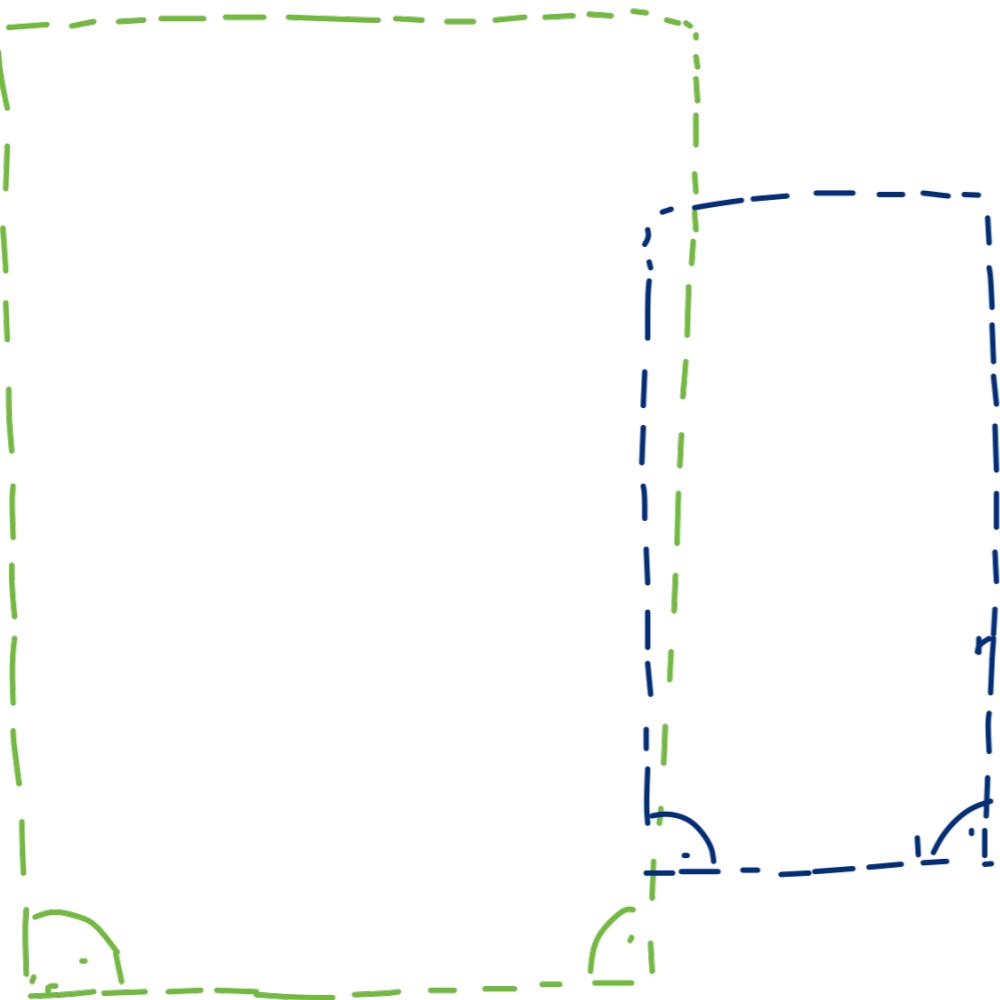
# "DIFFICULT" QUESTION: DO THESE POLYGONS INTERSECT?

---



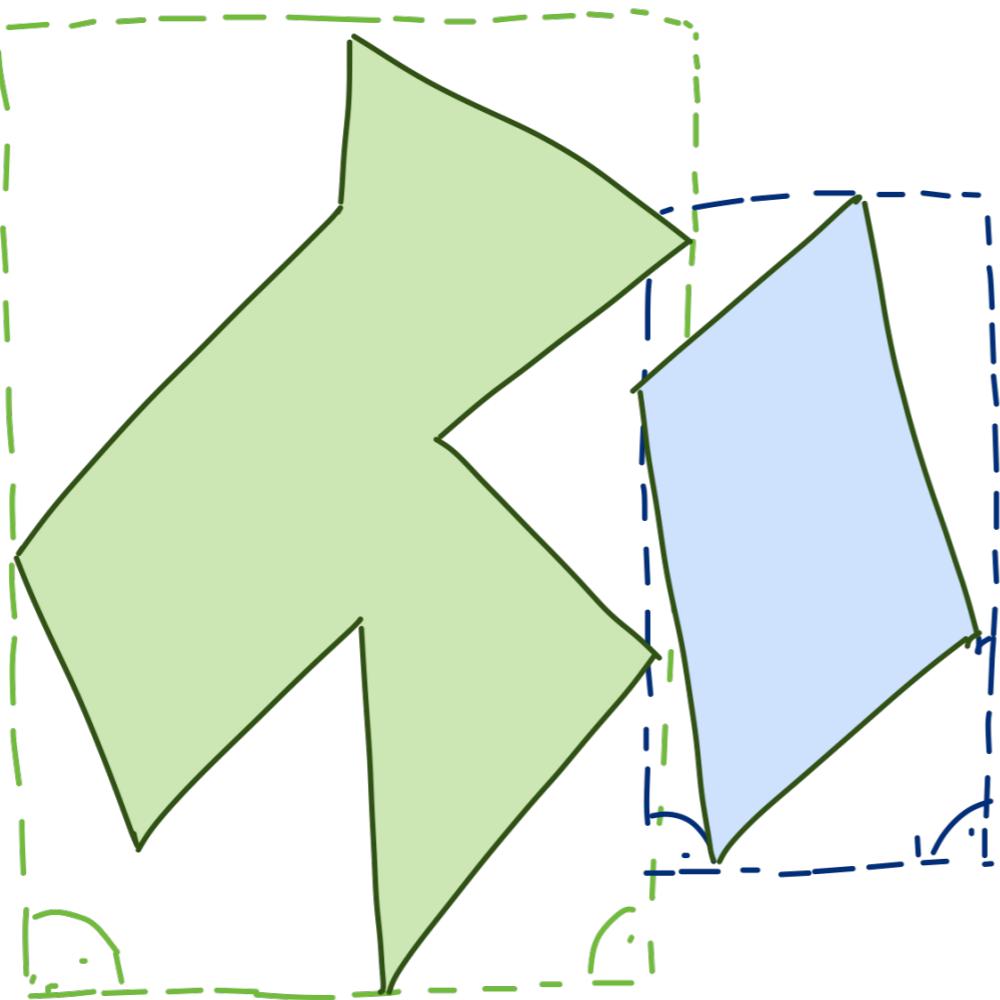
# "EASY" QUESTION: DO THESE RECTANGLES INTERSECT?

---



# DO THESE POLYGONS INTERSECT?

---



# INTERSECTION RULES

---

- Geometries don't intersect if their bounding boxes don't intersect
- Geometries **may** intersect if their bounding boxes intersect

# ST\_INTERSECTS OPTIMIZATION

---

```
(l.way && r.way) AND _st_intersects(l.way,  
r.way)
```

- Use the fast bbox intersection to rule out geometries which don't intersect.
- Verify the intersection using the slower but accurate method for the rest.

```
.....  
create index index_line on  
planet_osm_line using gist(way);
```

.....  
**Nested Loop** (cost=0.28..2662.01 rows=14974 width=13) (actual time=0.833..280.846 rows=3751 loops=1)

  -> **Seq Scan** on planet\_osm\_roads r (cost=0.00..145.45 rows=184 width=309) (actual time=0.051..0.478 rows=184 loops=1)

    Filter: (bridge = 'yes'::text)

    Rows Removed by Filter: 2572

  -> **Index Scan** using index\_line on planet\_osm\_line l (cost=0.28..13.67 rows=1 width=211) (actual time=0.414..1.521 rows=20 loops=184)

**Index Cond:** (way && r.way)

**Filter:** \_st\_intersects(way, r.way)

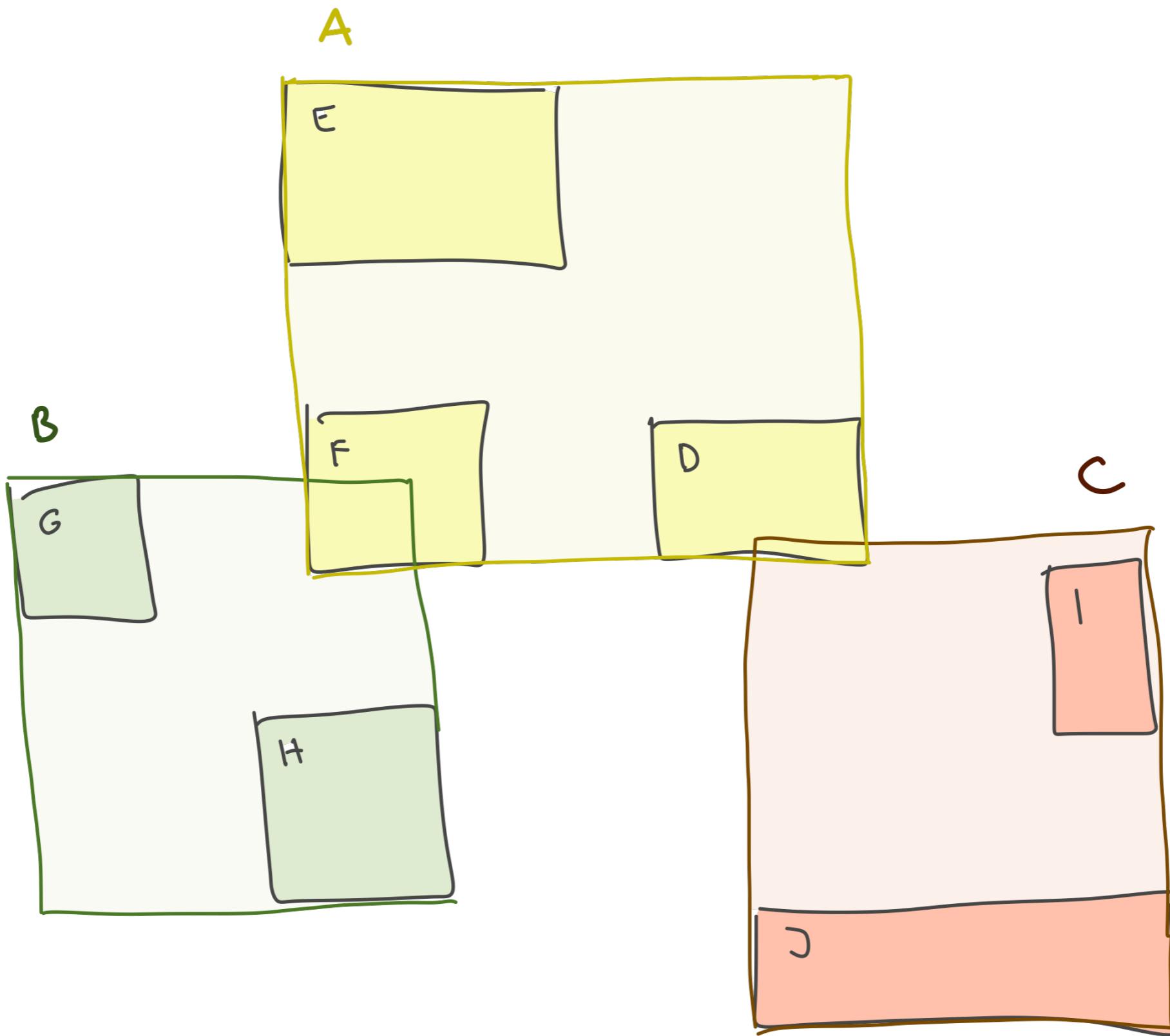
    Rows Removed by Filter: 61

Planning time: 0.413 ms

Execution time: 281.089 ms

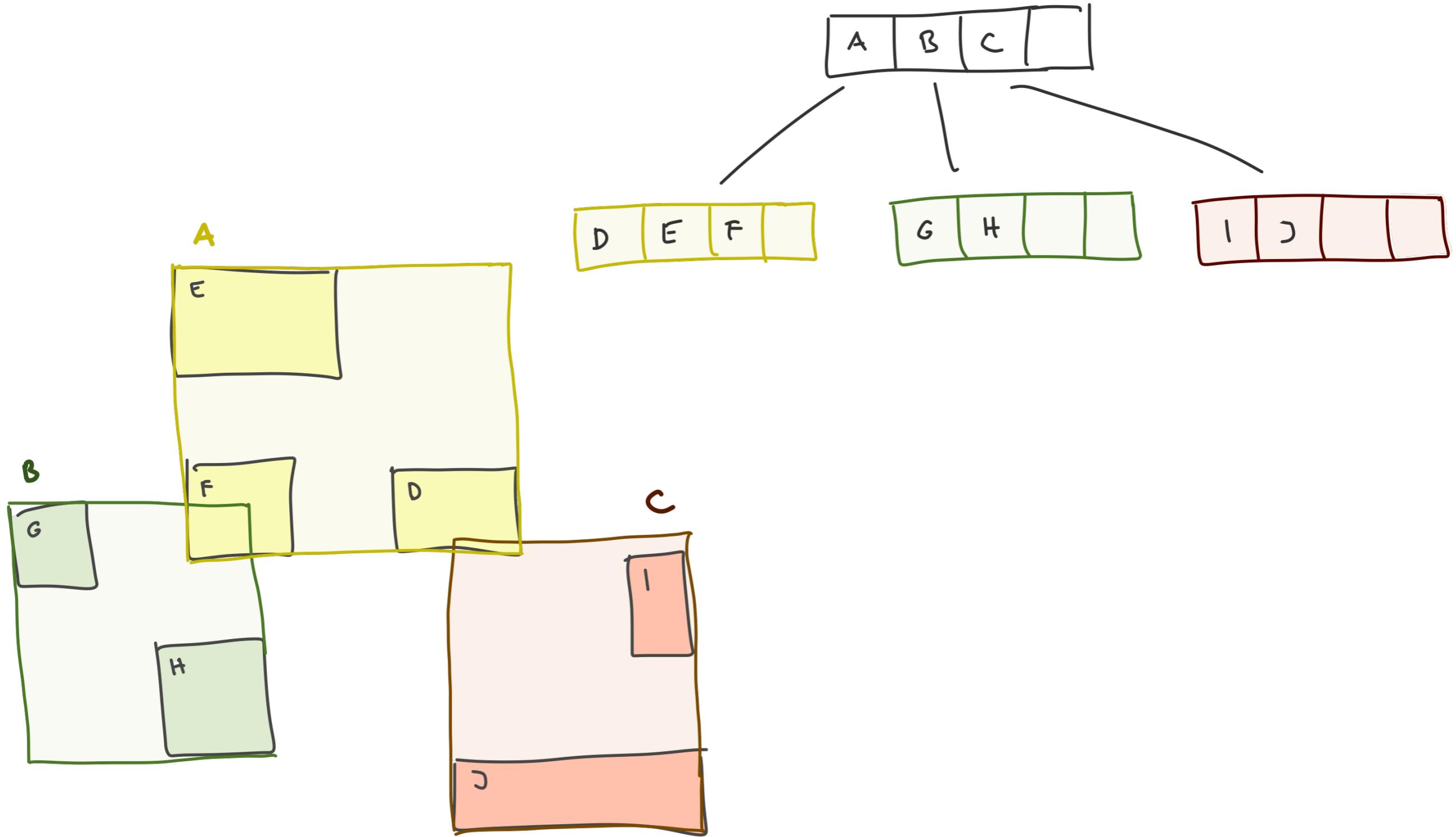
# R-TREE

---



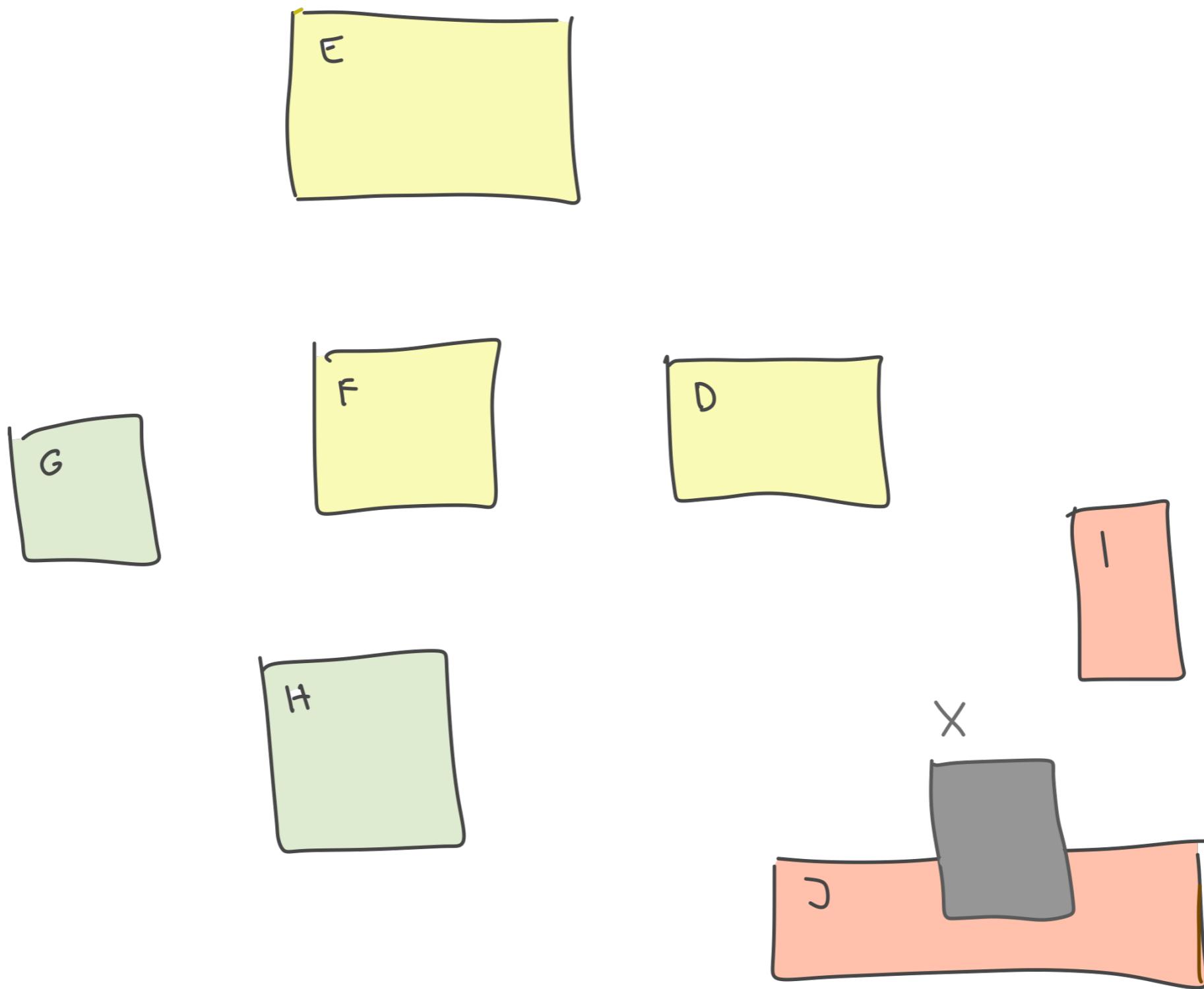
# R-TREE

---



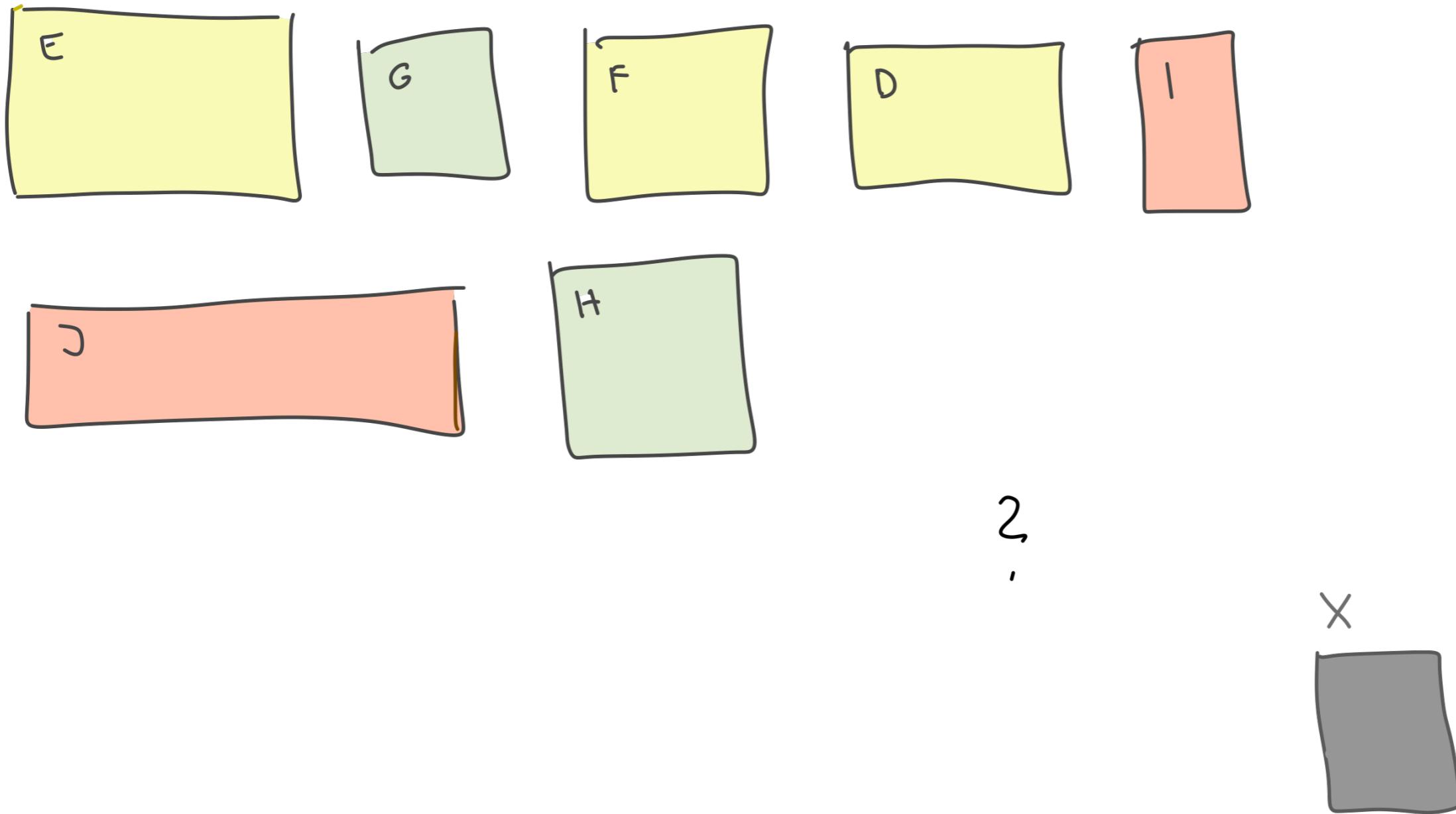
# R-TREE

---



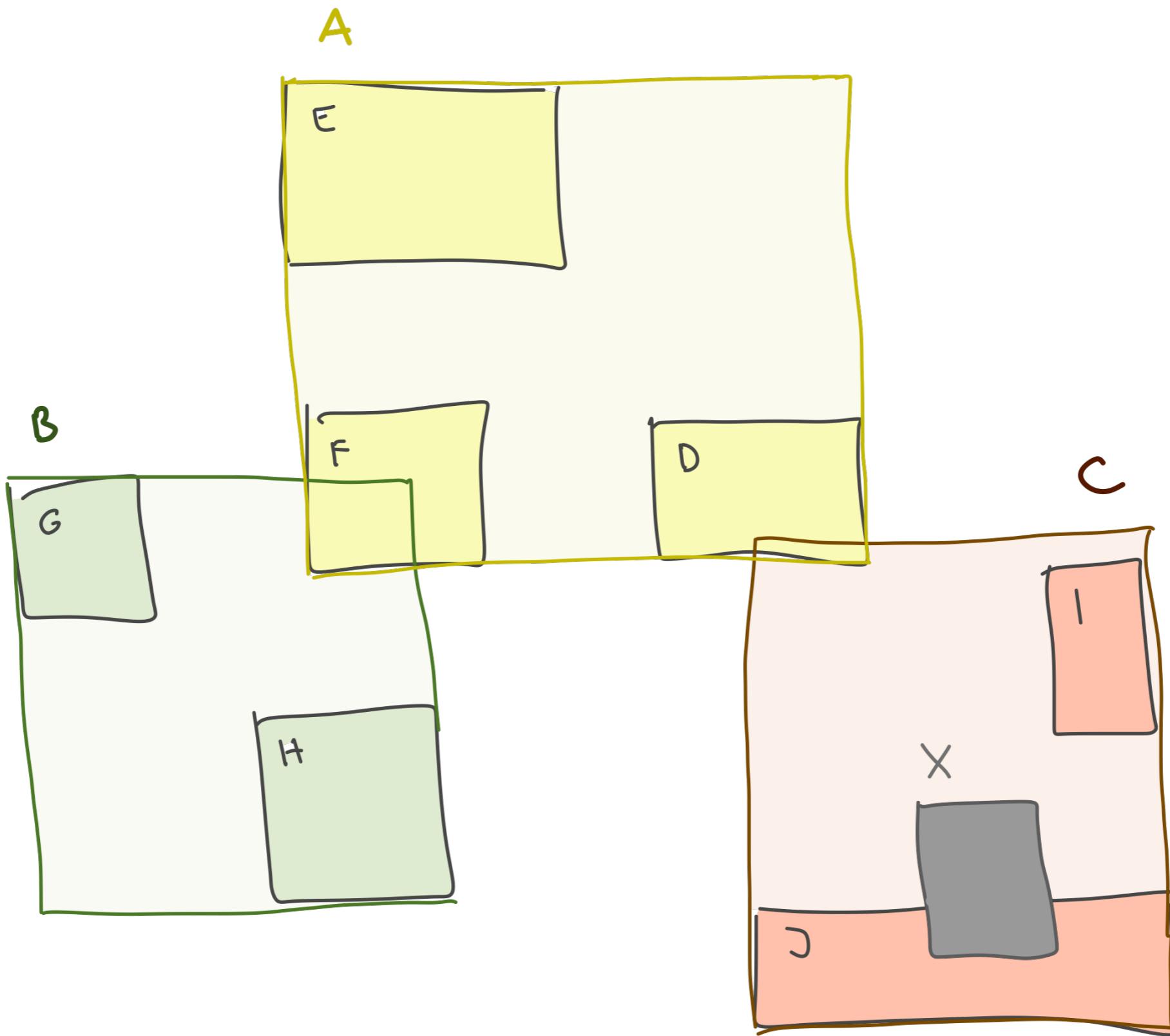
# R-TREE

---



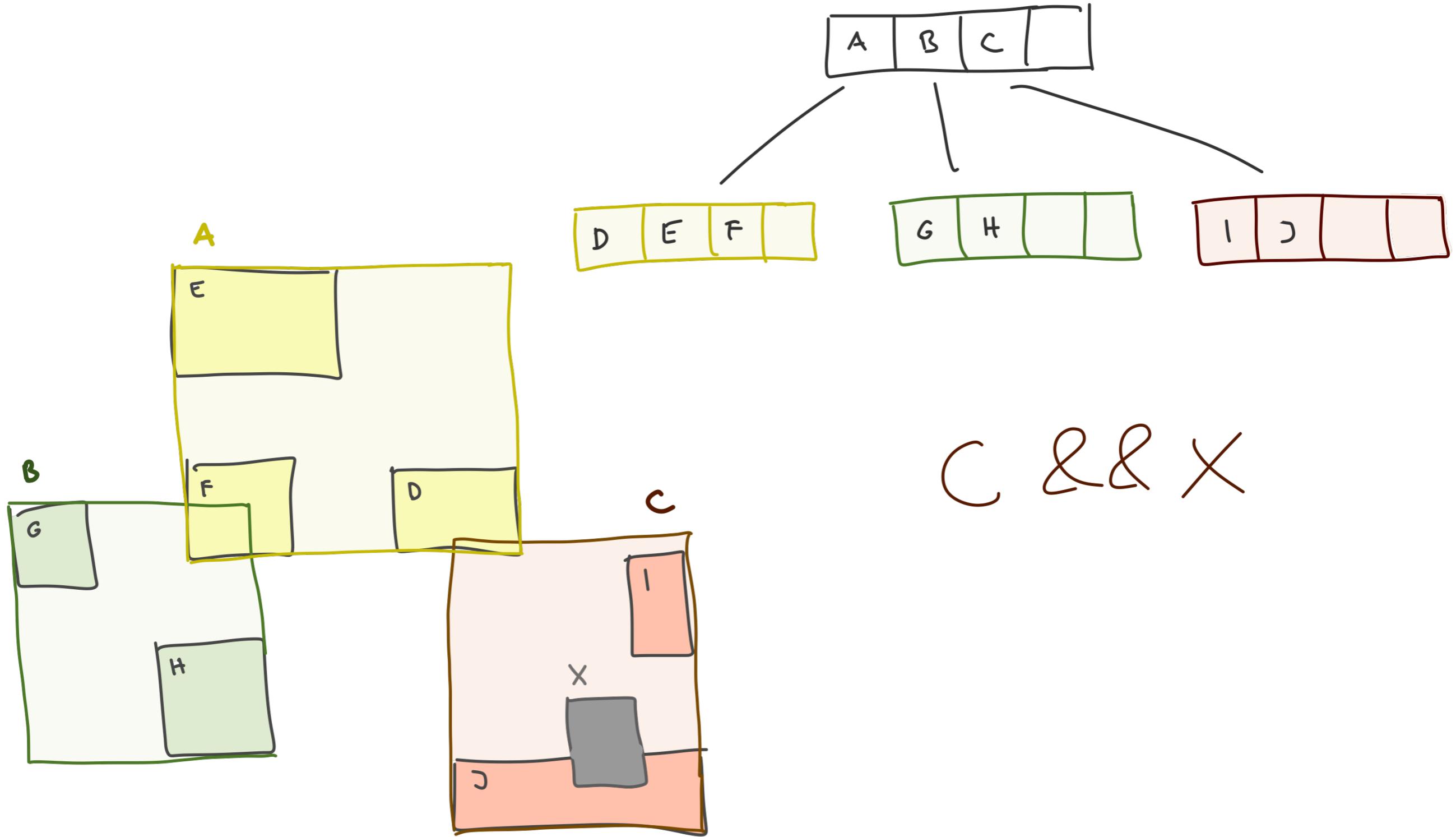
# R-TREE

---



# R-TREE

---



## ST\_DISTANCE/ST\_DWITHIN

---

where `st_distance(a.way, b.way) <= 1000`

vs.

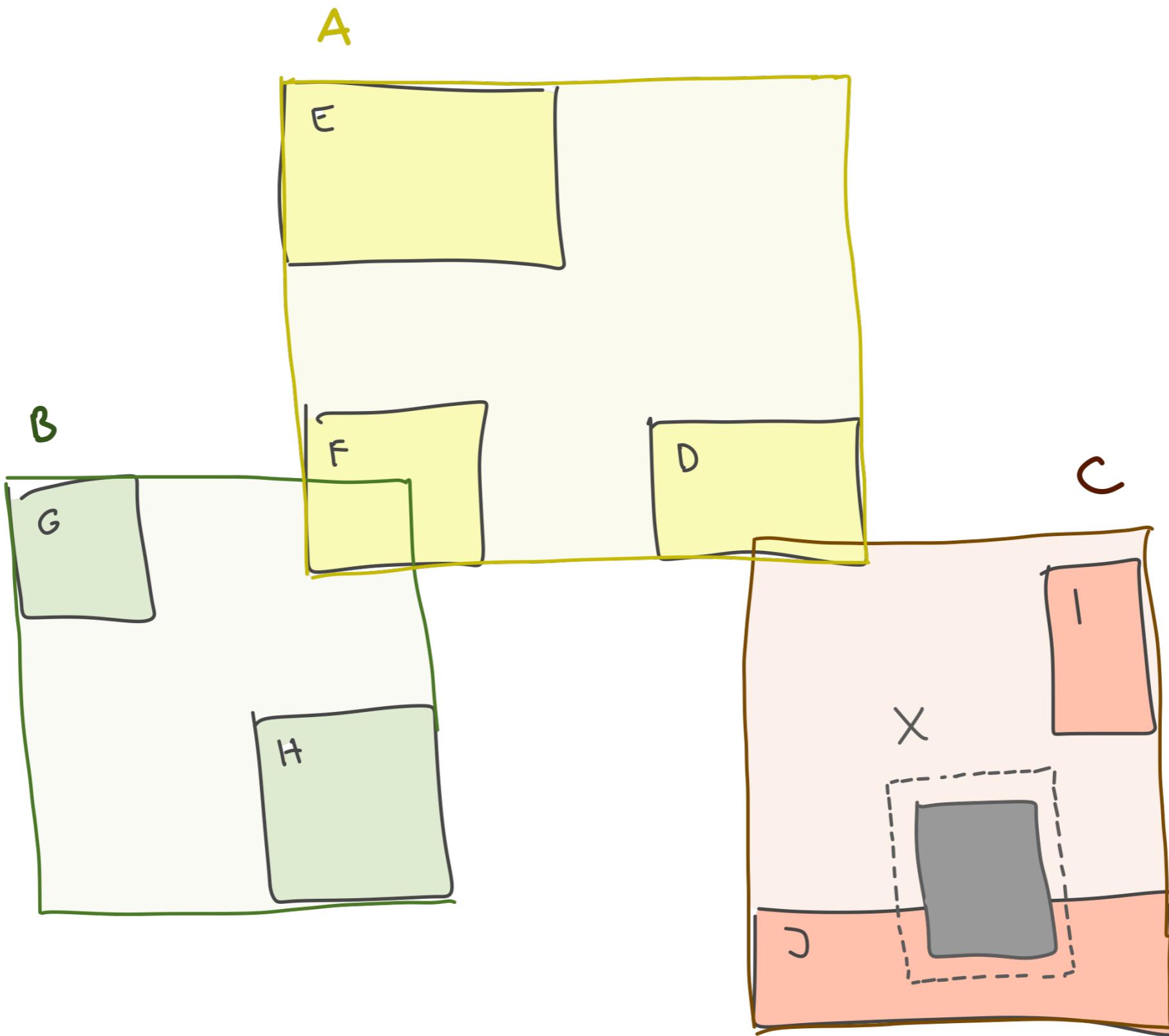
`where st_dwithin(a.way, b.way, 1000)`

```
.....  
explain analyze select *  
from planet_osm_polygon p  
join planet_osm_polygon p2 on  
st_dwithin(p.way, p2.way, 1000)  
where p.name = 'Fakulta informatiky a  
informačných technológií STU';
```

```
.....  
Nested Loop (cost=8.62..61.71 rows=1 width=2468) (actual  
time=10.813..128.344 rows=49606 loops=1)  
  -> Bitmap Heap Scan on planet_osm_polygon p (cost=4.31..12.11  
rows=2 width=1234) (actual time=0.038..0.038 rows=1 loops=1)  
      [...]  
  -> Bitmap Heap Scan on planet_osm_polygon p2 (cost=4.32..24.79  
rows=1 width=1234) (actual time=10.768..114.630 rows=49606 loops=1)  
      Recheck Cond: (way && st_expand(p.way, 1000::double precision))  
      Filter: ((p.way && st_expand(way, 1000::double precision)) AND  
      _st_dwithin(p.way, way, 1000::double precision))  
      Heap Blocks: exact=1469  
  -> Bitmap Index Scan on index_polygon (cost=0.00..4.32 rows=5  
width=0) (actual time=9.196..9.196 rows=49606 loops=1)  
      Index Cond: (way && st_expand(p.way, 1000::double  
precision))  
Planning time: 0.478 ms  
Execution time: 129.885 ms
```

# ST\_EXPAND

---



# WHAT YOU SHOULD KNOW

---

- Principles of R-trees (not how they are created and balanced, but how they are searched and why they make intersection search faster)
- Difference between `st_distance(geom1, geom2) <= 1000` and `st_dwithin(geom1, geom2, 1000)`