Introduction to Urban Data Science
Lecture 1

Spatial Data Processing

Harish Doraiswamy
New York University

NYU

VIDA VISUALIZATION IMAGING AND DATA ANALYSIS CENTER

Outline

- Spatial Data
- Spatial Queries
- Spatial Indexes

Spatial Data

Infrastructure

Environment

Photo by [MTA]

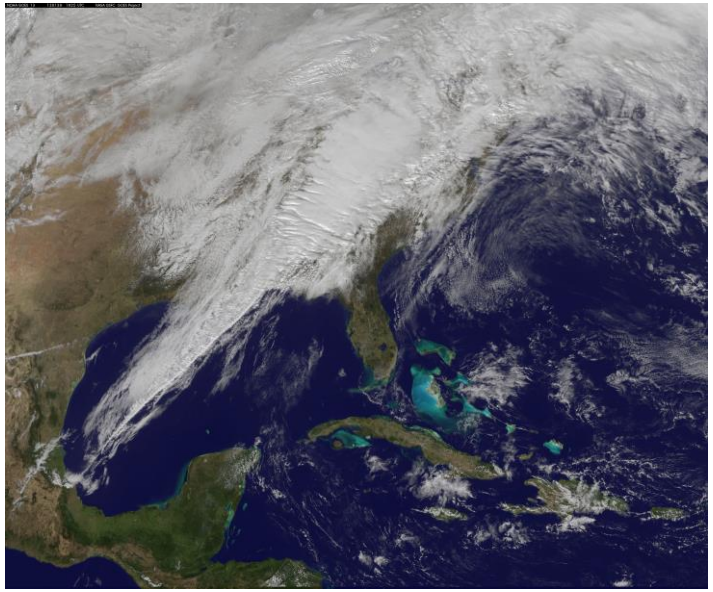
Photo by *Yinka Oyesiku*

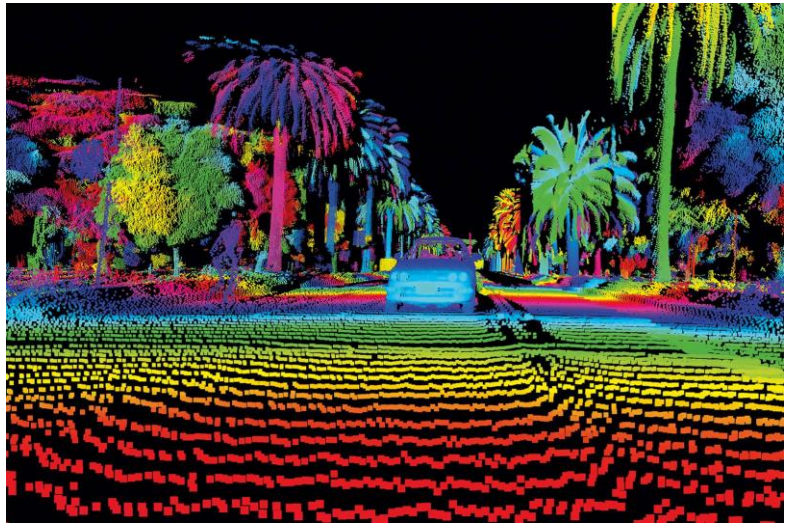
People

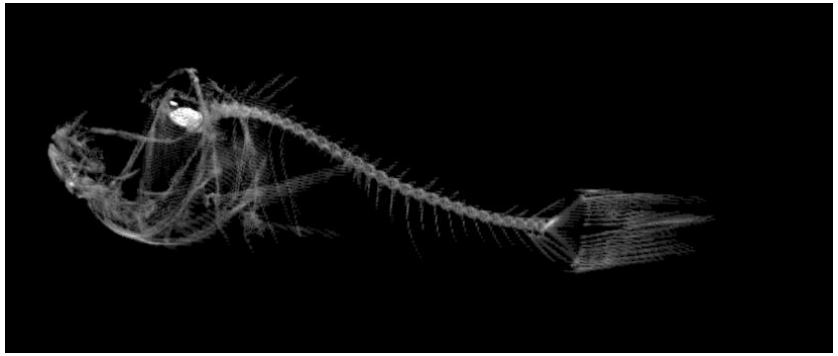
flickr

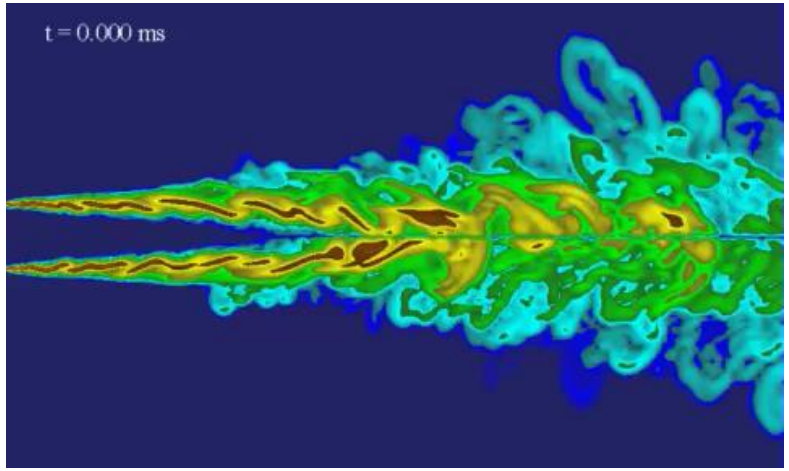
twitter

Spatial Data









Spatial Data

- Spatial Attributes
  - 2D – (x,y)
  - 3D – (x,y,z)
- Spatial Data Primitives
  - Points
  - Lines
  - Polygons
- Other Attributes
  - Time
  - Everything else

Spatial Queries

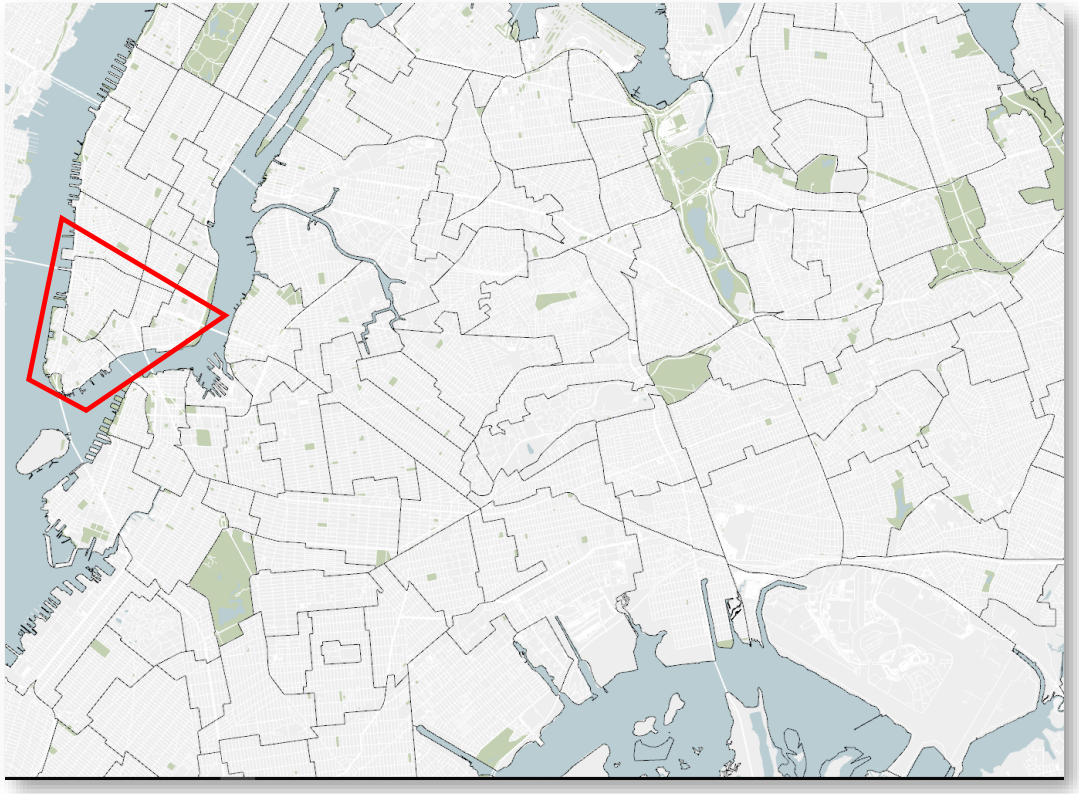
Spatial Selection Queries

- Select spatial objects that satisfy a spatial constraint
- Spatial Constraint
  - Intersection
  - Spatial Distance

Spatial Selection Queries

- Select spatial objects that satisfy a spatial constraint

SELECT *
FROM taxi *T*
WHERE *T*.pickup inside *LowerManhattan*



Spatial Selection Queries

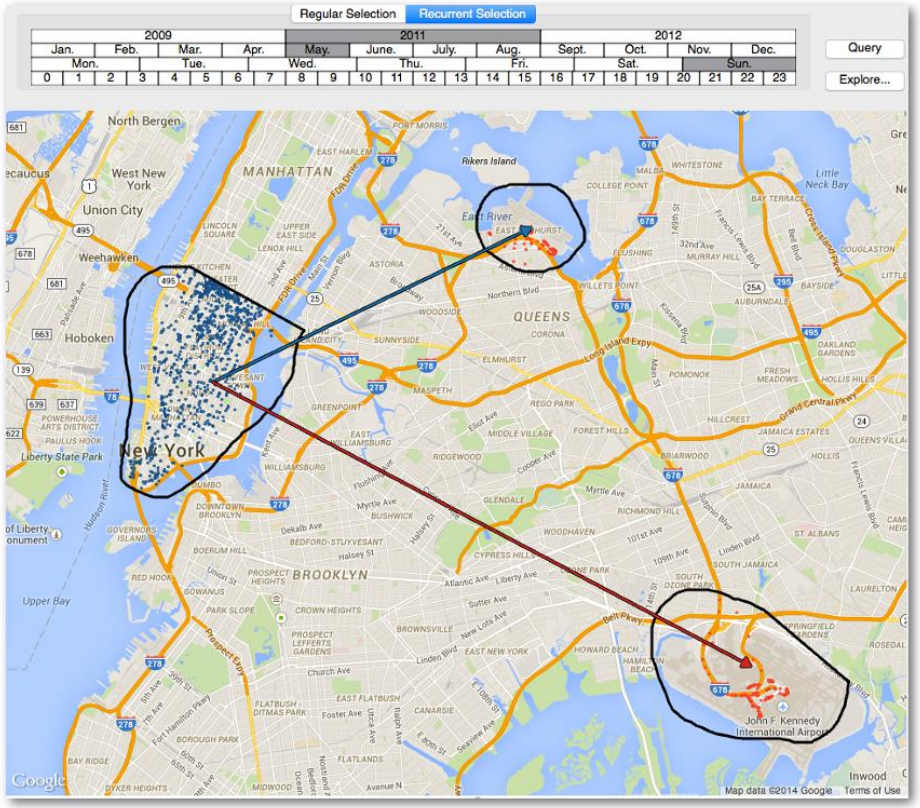
- Select spatial objects that satisfy a spatial constraint

SELECT *
FROM taxi *T*
WHERE *T*.pickup inside *LowerManhattan*
AND
(*T*.dropoff inside *JFK* OR *T*.dropoff inside *LGA*)

Spatial Selection Queries

- Select spatial objects that satisfy a spatial constraint

SELECT *
FROM taxi *T*
WHERE *T*.pickup inside *LowerManhattan*
AND
(*T*.dropoff inside *JFK* OR *T*.dropoff inside *LGA*)
AND *T*.picktime in *May 2011*

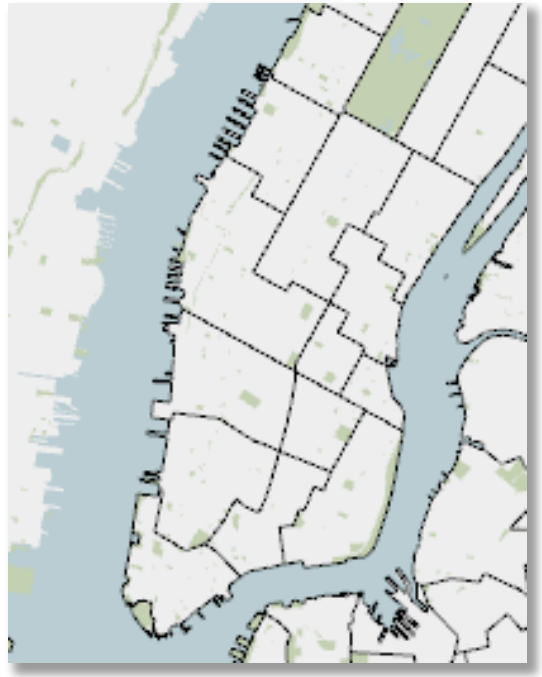


TaxiVis [IEEE TVCG 2013]

Spatial Joins

- Select pairs of spatial objects satisfying a spatial constraint

SELECT *
FROM taxi *T*, neighborhoods *N*
WHERE *T*.pickup inside *N*.polygon

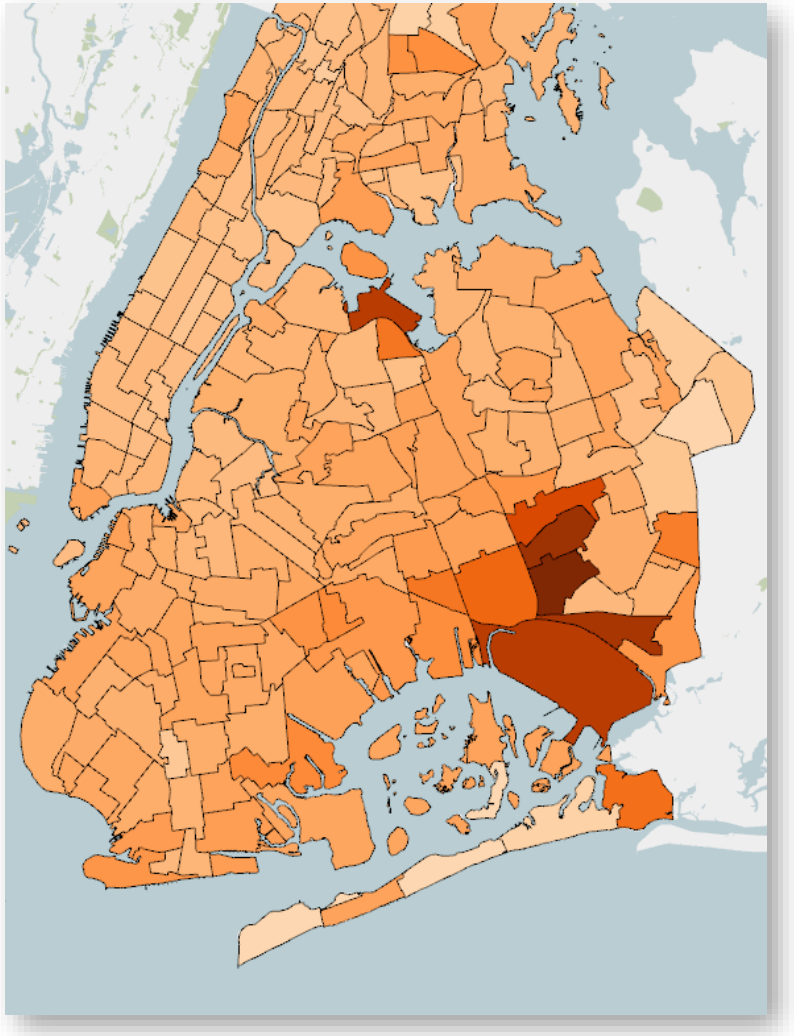


Spatial Aggregation Queries

```
SELECT COUNT(*)
FROM taxi T, neighborhoods N
WHERE T.pickup INSIDE N.geometry
GROUP BY N.id
```

Spatial Aggregation Queries

```
SELECT AVG(T.duration)
FROM taxi T, neighborhoods N
WHERE T.pickup INSIDE N.geometry
GROUP BY N.id
```

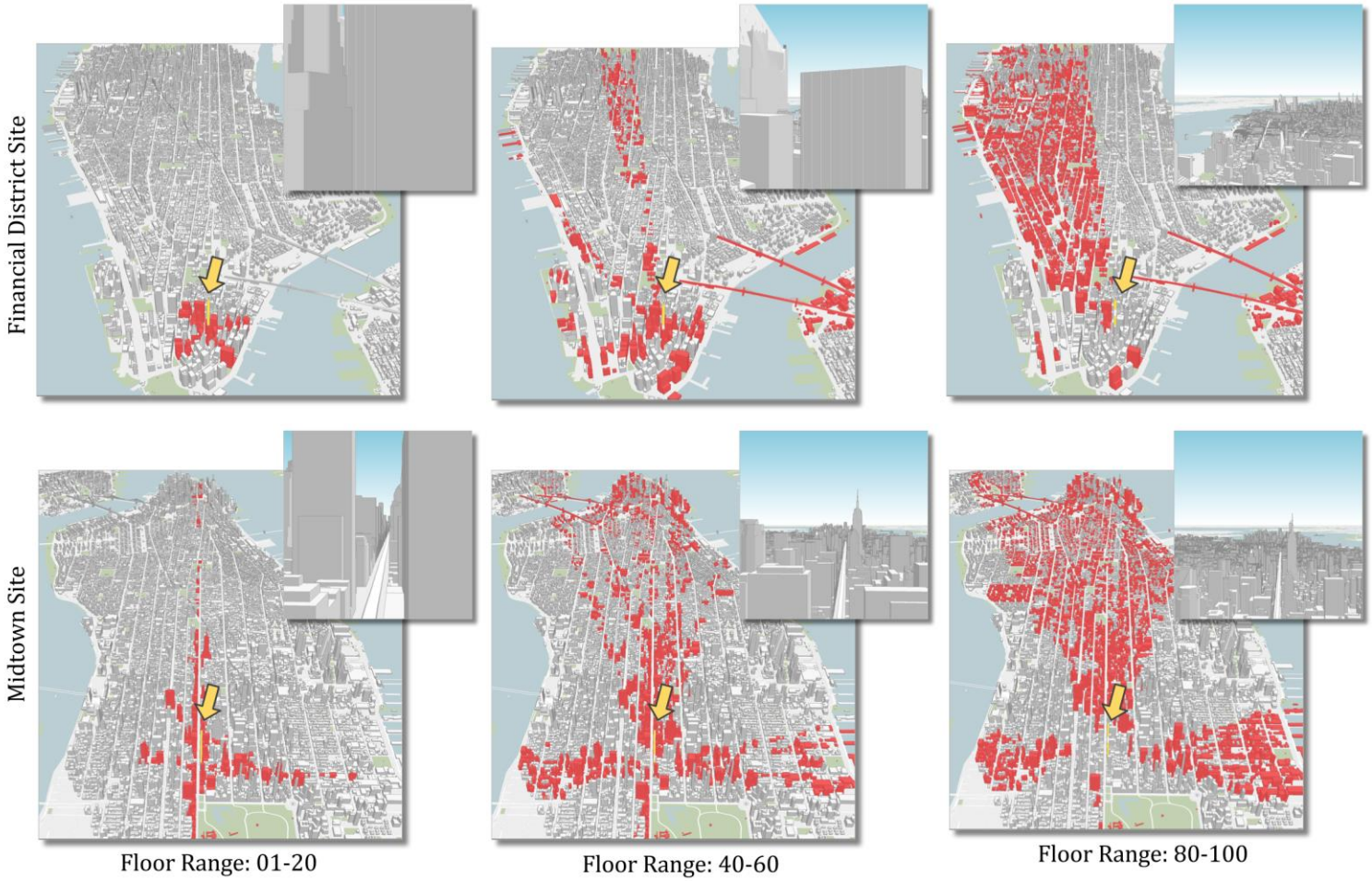


Nearest Neighbor Queries

SELECT TOP(10)
FROM building-lots *B*, crime *C*
ORDER BY DISTANCE(*B*.geometry, *C*.location)

Geometric Queries

- Voronoi regions
- Skyline
- Convex hulls
- Views

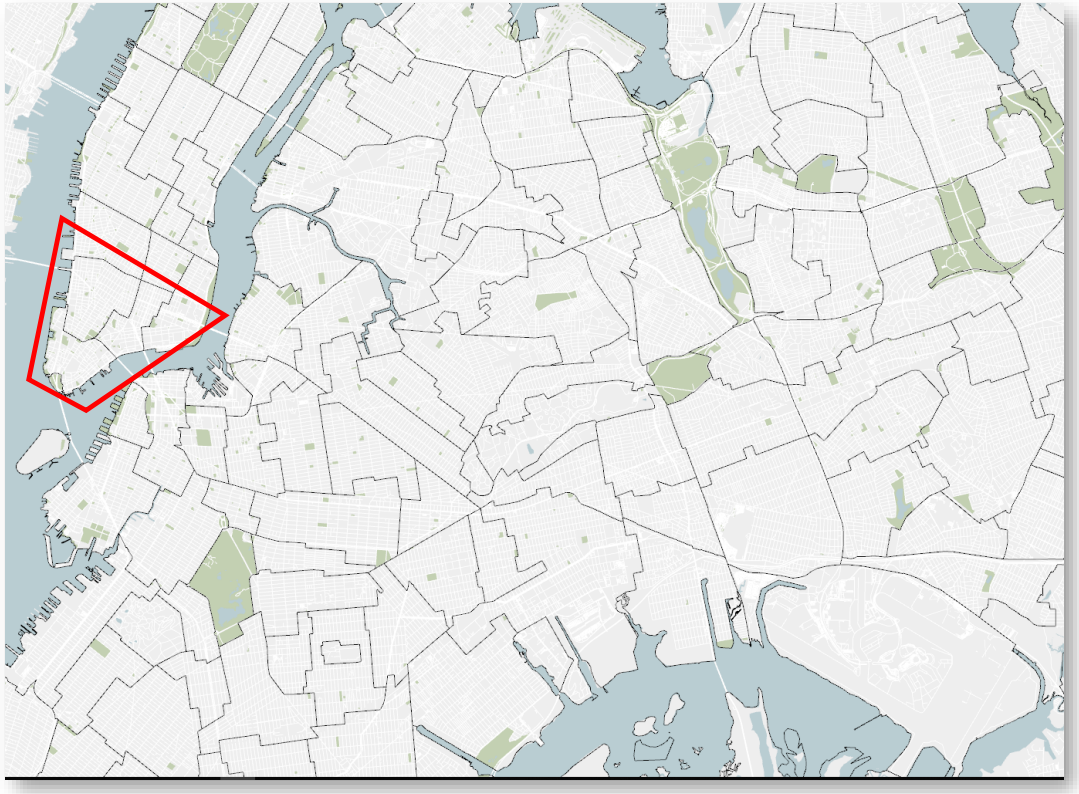


Topology-based Catalogue Exploration Framework for Identifying View-Enhanced Tower Designs
Harish Doraiswamy, Nivan Ferreira, Marcos Lage, Huy T. Vo, Luc Wilson, Heidi Werner, Muchan Park, Claudio Silva
ACM Transactions on Graphics (SIGGRAPH Asia '15), 34(6), 2015, 230:1-230:13

Spatial Selection Queries

- Select spatial objects that satisfy a spatial constraint

SELECT *
FROM taxi *T*
WHERE *T*.pickup inside *LowerManhattan*

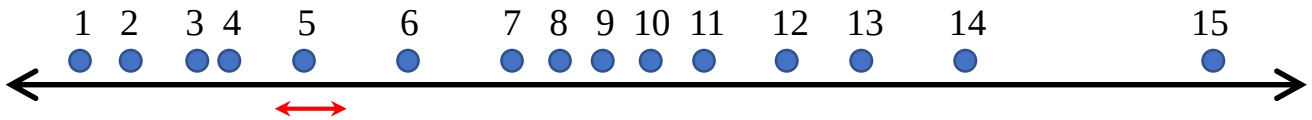


Spatial Index

1-Dimensional Example

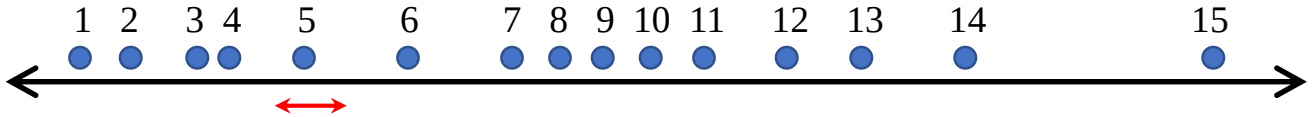
Find points between x and y

1-Dimensional Example

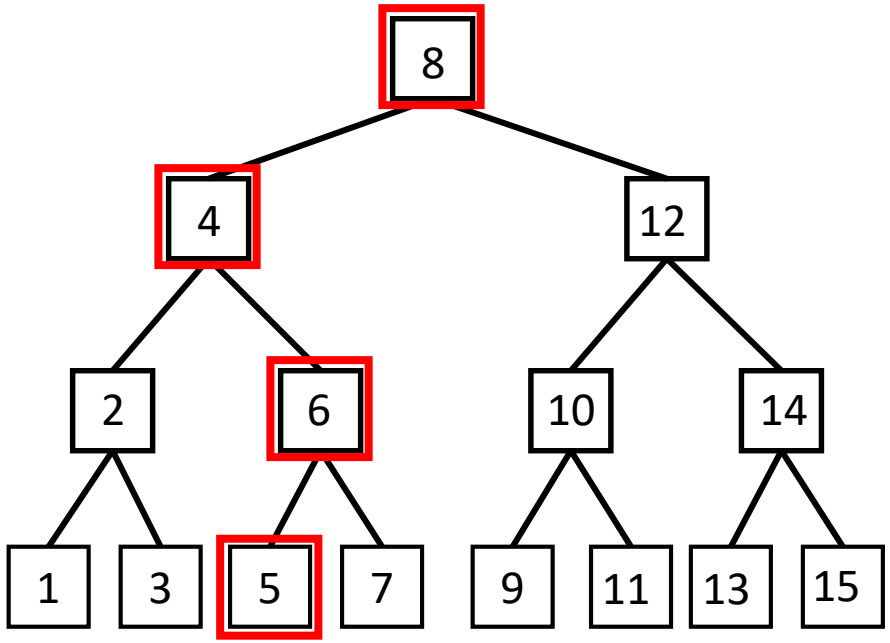


Find points with value 5

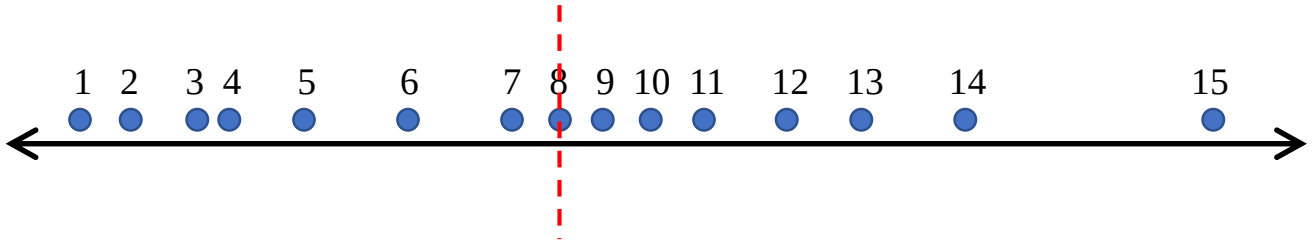
1-Dimensional Example



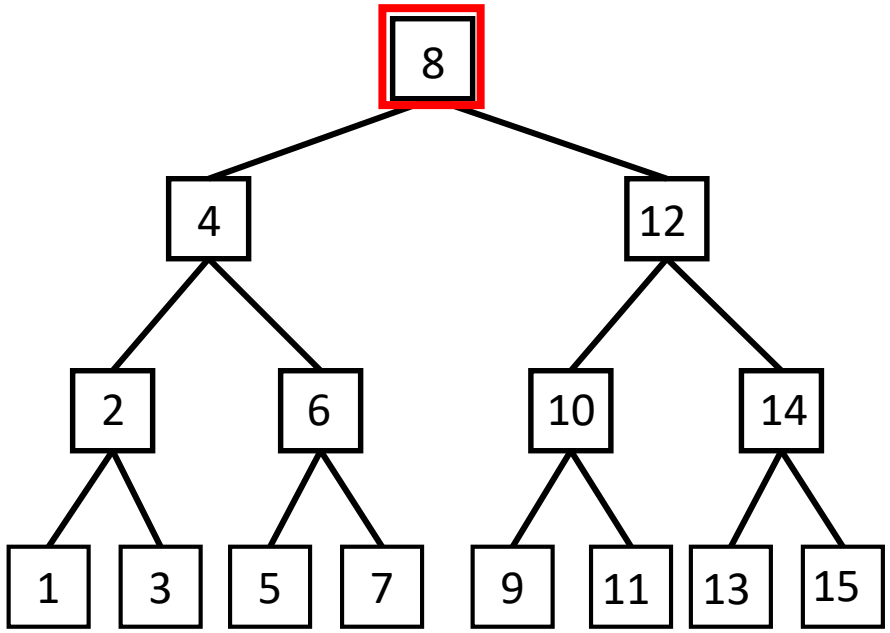
Binary Search Tree



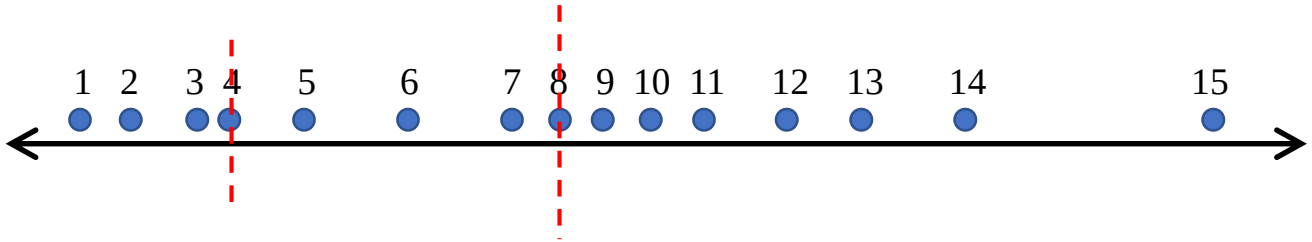
1-Dimensional Example



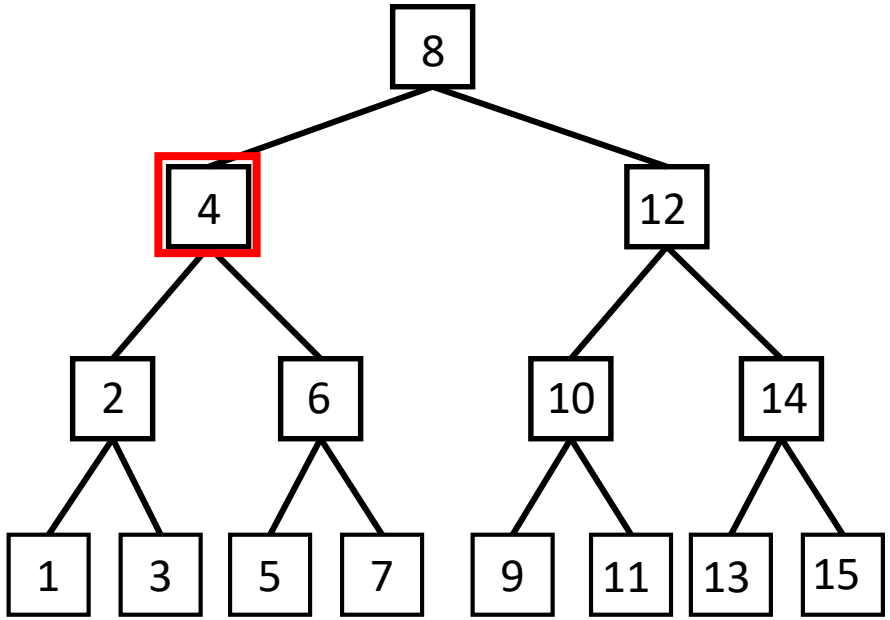
Binary Search Tree



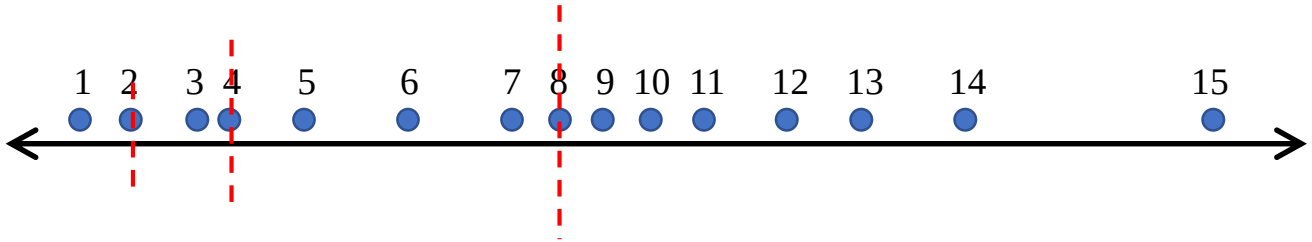
1-Dimensional Example



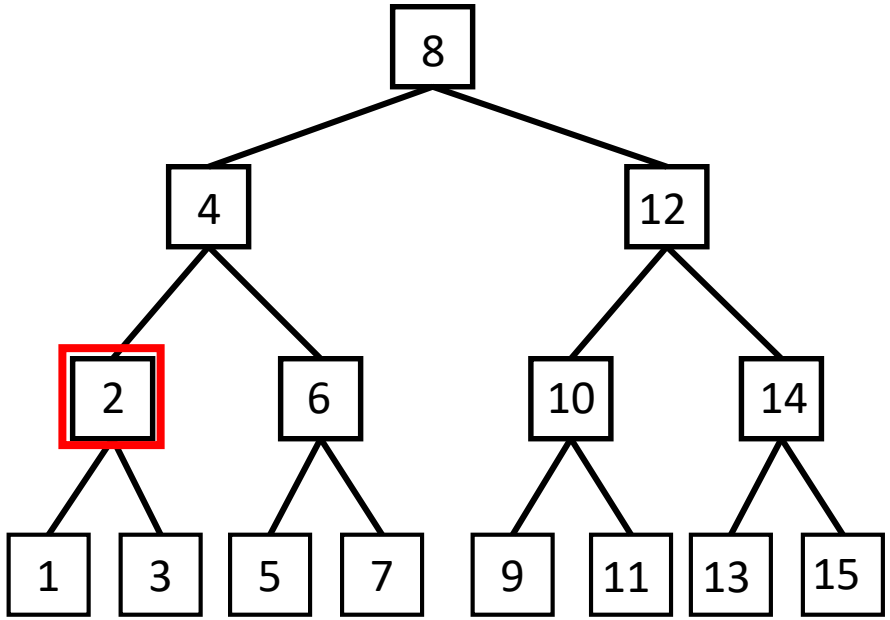
Binary Search Tree



1-Dimensional Example



Binary Search Tree

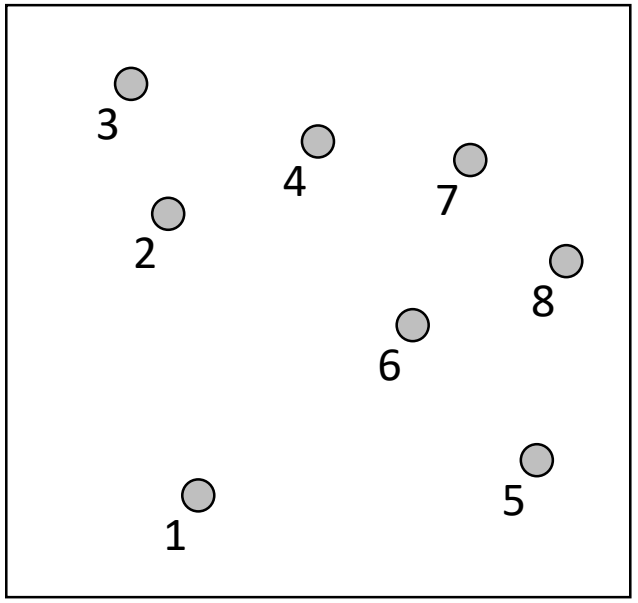


KD-Tree

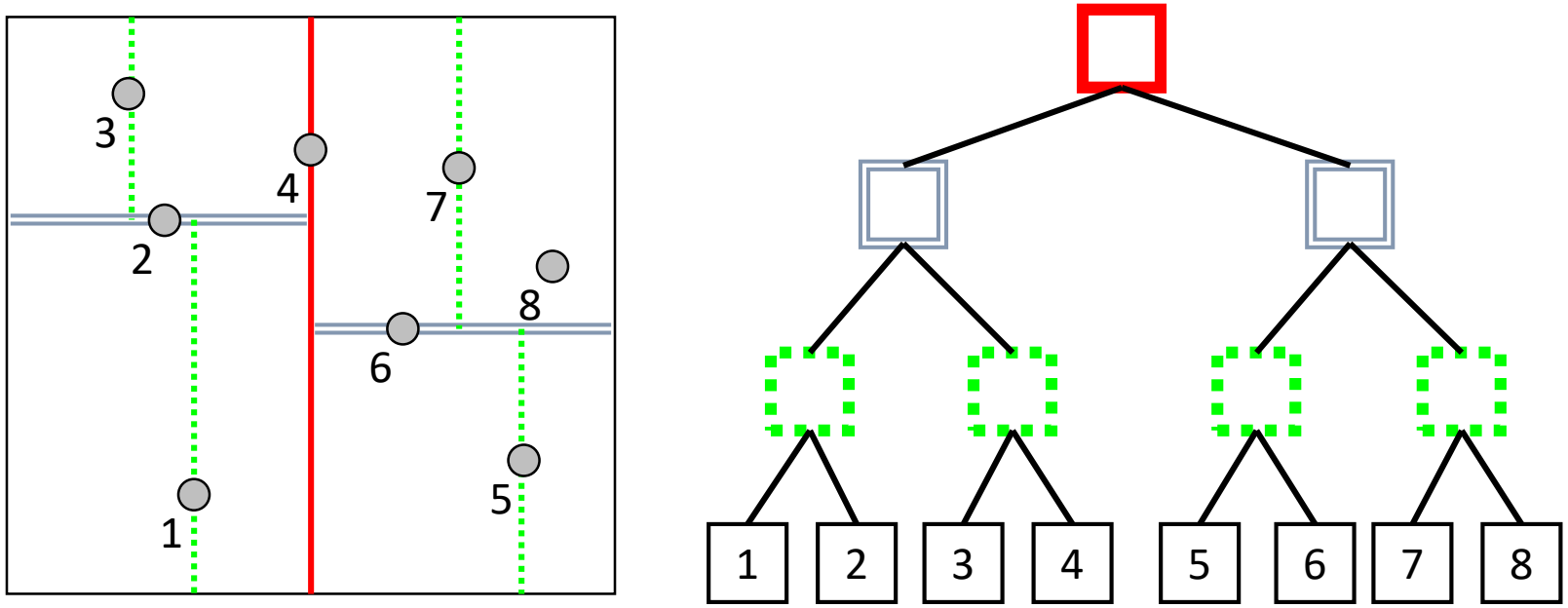
- Extension of a Binary Search Tree
- Supports k-dimensional tree
- Focus on 2D

KD-Tree

- At each level split with respect to one of the dimensions

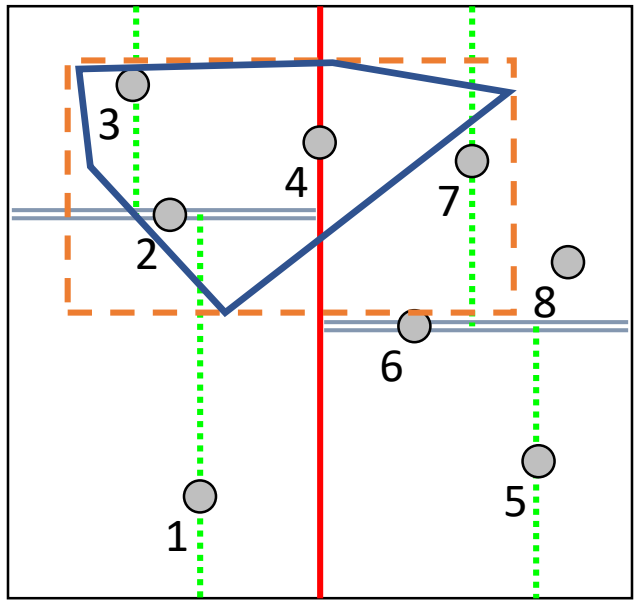


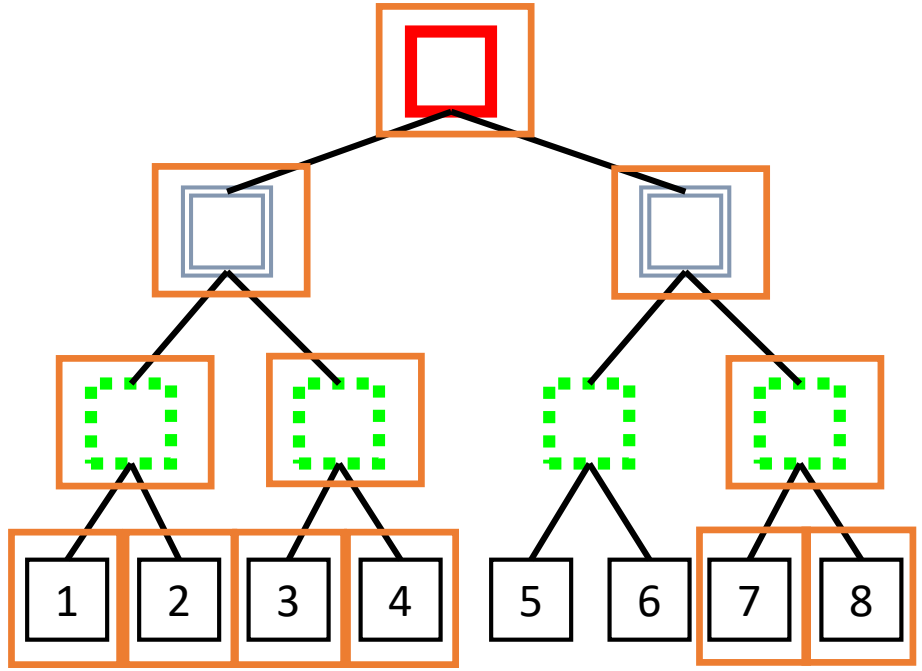
KD-Tree



KD-Tree

- Polygon containment query
  - Search based on Bounding Box
  - Test with query polygon





KD-Tree

- Advantages
  - Handles skewed data
  - Can be extended to any dimension
- Disadvantages
  - Adding new data need not create balanced tree

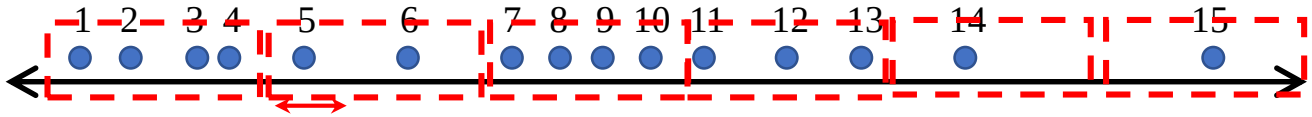
Traditional Index Analogy

- Binary Tree

Traditional Index Analogy

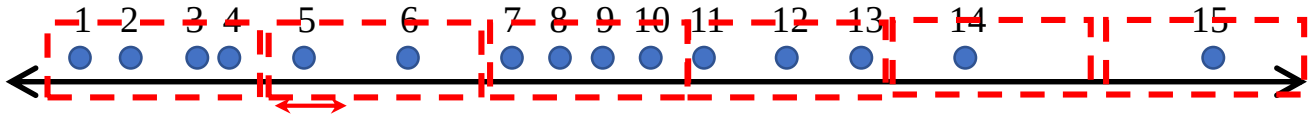
- Binary Tree → KD-Tree
- Hash Index

1-Dimensional Example



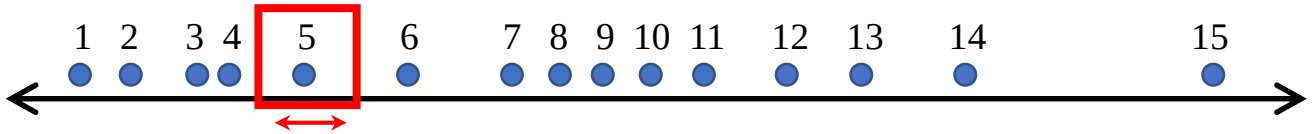
- Create bins using a hash function

1-Dimensional Example



- Create bins using a hash function
- Query
  - Identify bin(s) satisfying query constraint

1-Dimensional Example

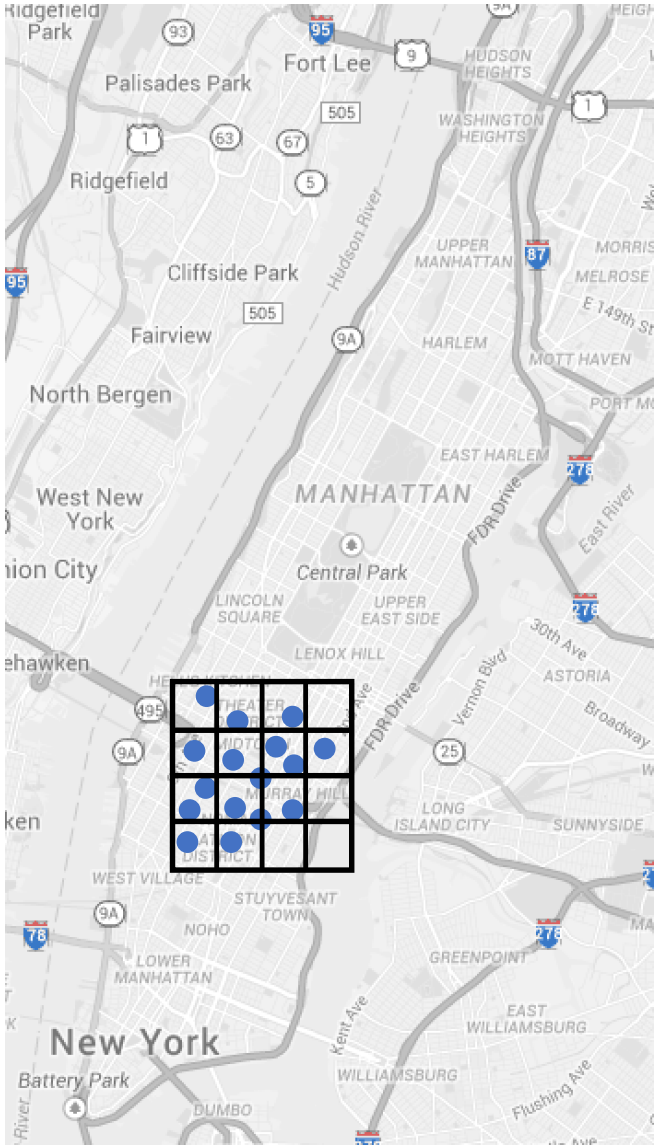


- Create bins using a hash function
- Query
  - Identify bin(s) satisfying query constraint
  - Search within the bin

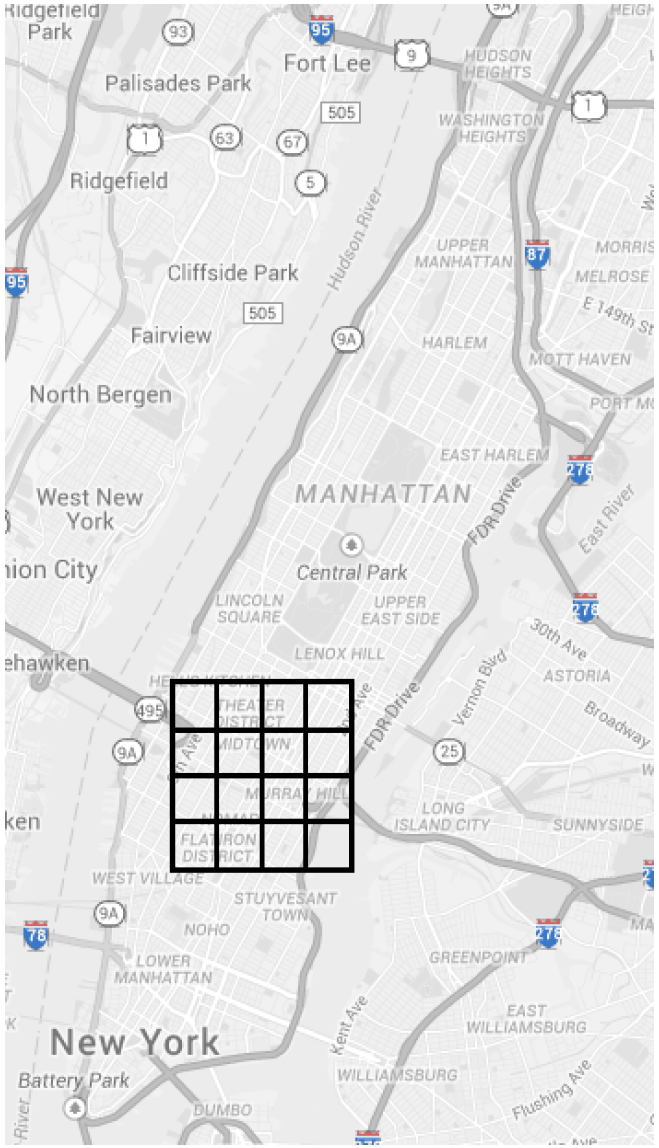
Grid Index

- Extension of hash index to higher dimensions
- Hash function is defined by a grid
- Idea
  - Overlay a grid covering the spatial region
  - Assign objects to different grid cells.

Grid Index


Ridgefield Park
Palisades Park
Fort Lee
Ridgefield
Cliffside Park
Fairview
North Bergen
West New York
Central Park
MANHATTAN
EAST HARLEM
HARLEM
UPPER MANHATTAN
HUDSON HEIGHTS
WASHINGTON HEIGHTS
MELROSE
MOTT HAVEN
LINCOLN SQUARE
UPPER EAST SIDE
LENOX HILL
ASTORIA
LONG ISLAND CITY
SUNNYSIDE
WEST VILLAGE
STUYVESANT TOWN
NOHO
LOWER MANHATTAN
GREENPOINT
EAST WILLIAMSBURG
WILLIAMSBURG
DUMBO
Battery Park
New York
Hudson River
East River
FDR Drive
Vernon Blvd
30th Ave
Broadway
Kent Ave
Flushing Ave

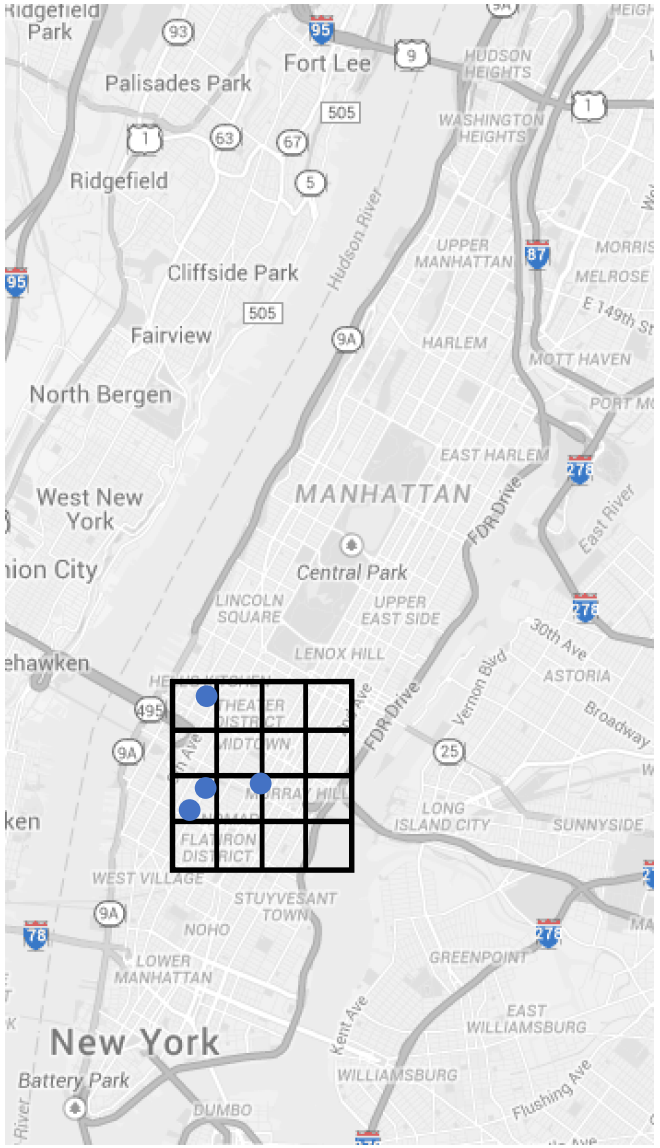
Grid Index


Palisades Park
Fort Lee
Ridgefield
Cliffside Park
Fairview
North Bergen
West New York
MANHATTAN
Central Park
New York
Battery Park
Hudson River
East River
FDR Drive
HARLEM
EAST HARLEM
UPPER MANHATTAN
LOWER MANHATTAN
LONG ISLAND CITY
ASTORIA
SUNNYSIDE
WILLIAMSBURG
GREENPOINT
DUMBO
NOHO
STUYVESANT TOWN
WEST VILLAGE
LENOX HILL
UPPER EAST SIDE
LINCOLN SQUARE
MOTT HAVEN
MELROSE
HUDSON HEIGHTS
WASHINGTON HEIGHTS

Grid Index

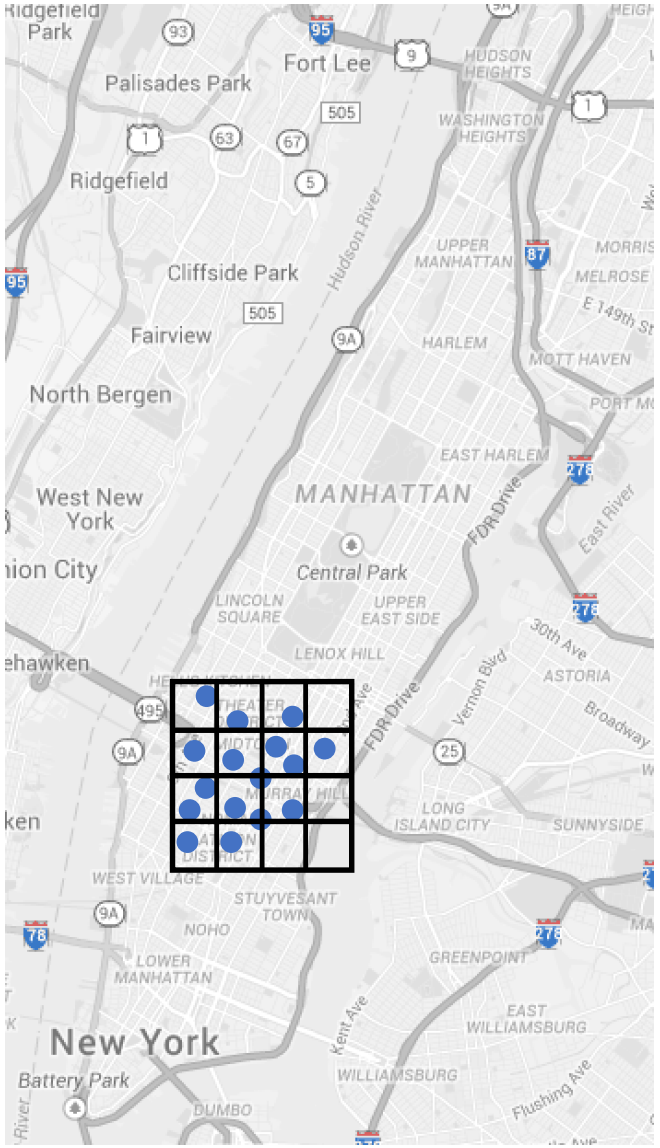

12 13 14 15
8 9 10 11
4 5 6 7
0 1 2 3
Fort Lee
Palisades Park
Ridgefield
Cliffside Park
Fairview
North Bergen
West New York
MANHATTAN
Central Park
New York
Battery Park
Hudson River
East River
FDR Drive

Grid Index



0	
1	
2	
3	
4	3 4
5	10
6	10
7	
8	
9	
10	
11	
12	1
13	
14	
15	

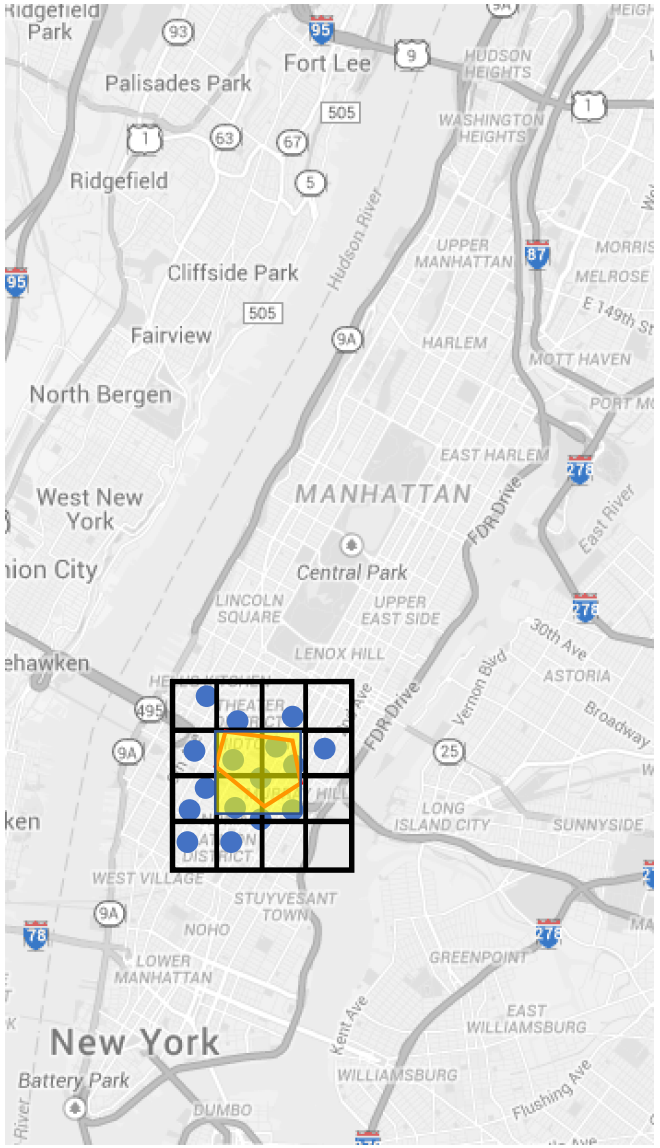
Grid Index



0	5
1	9 11
2	11
3	
4	3 4
5	8 10 11
6	10 11 15
7	
8	2
9	7
10	12 14
11	16
12	1
13	6
14	14
15	

Grid Index

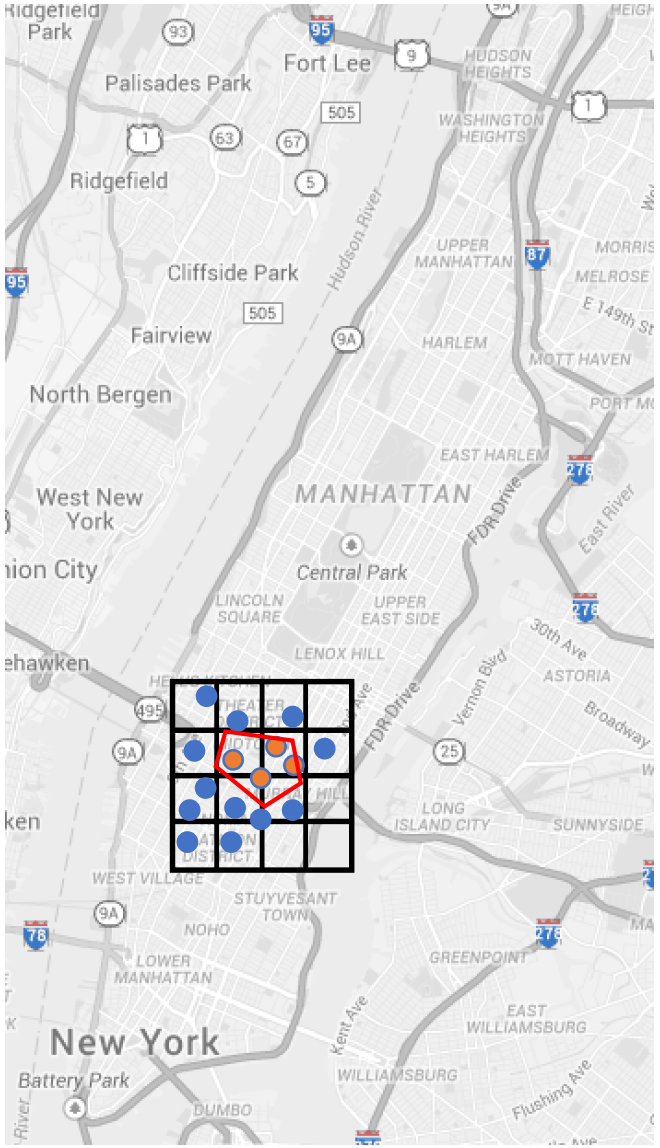
- Find Cells Intersected
  - 5,6,9,10
- Test all points in these cells



0	5
1	9 11
2	11
3	
4	3 4
5	8 10 11
6	10 11 15
7	
8	2
9	7
10	12 14
11	16
12	1
13	6
14	14
15	

Grid Index

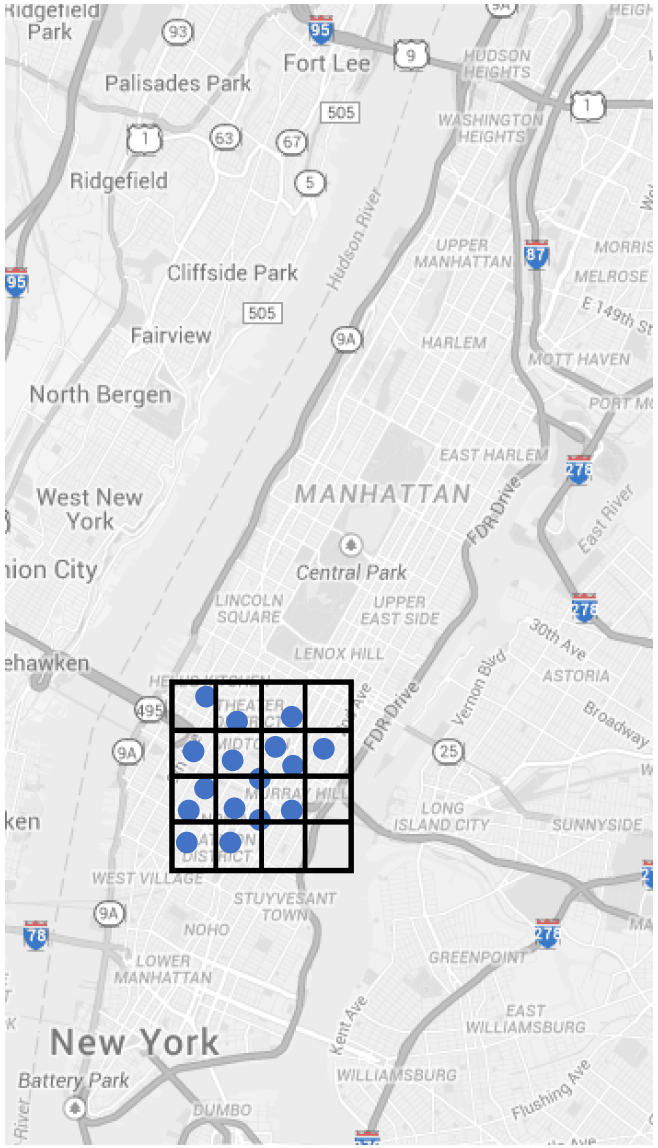
- Find Cells Intersected
  - 5,6,9,10
- Test all points in these cells



0	5
1	9 11
2	11
3	
4	3 4
5	8 10 11
6	10 11 15
7	
8	2
9	7
10	12 14
11	16
12	1
13	6
14	14
15	

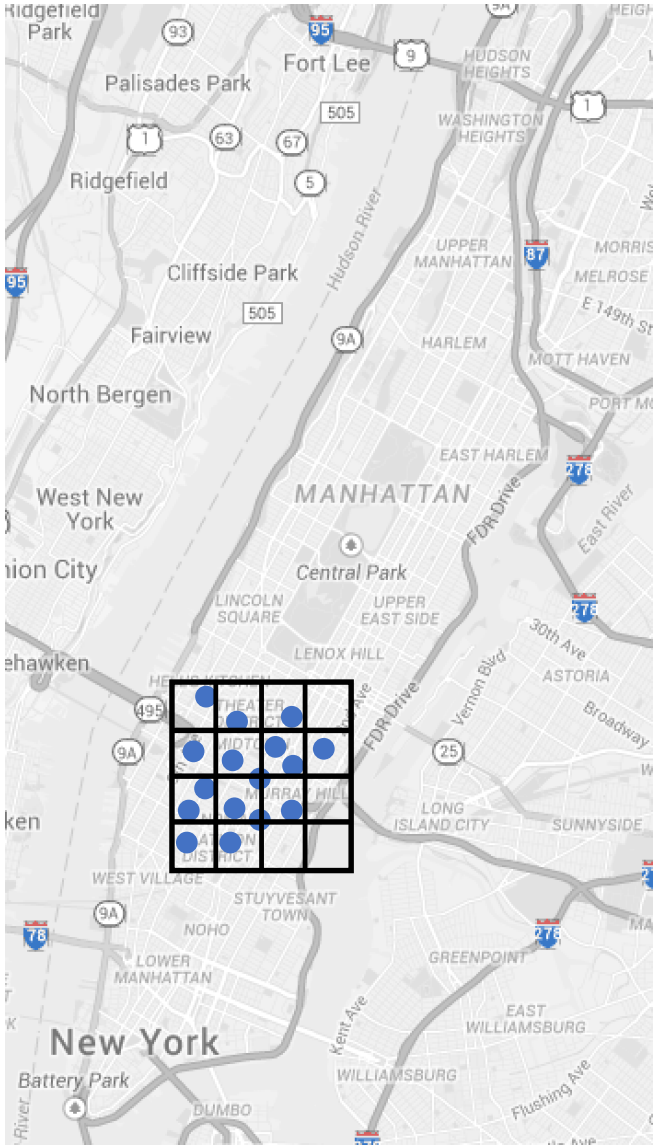
Grid Index

- Advantages
  - Simple
  - Adding new data



Grid Index

- Disadvantages
  - Grid Size?
  - Data skew

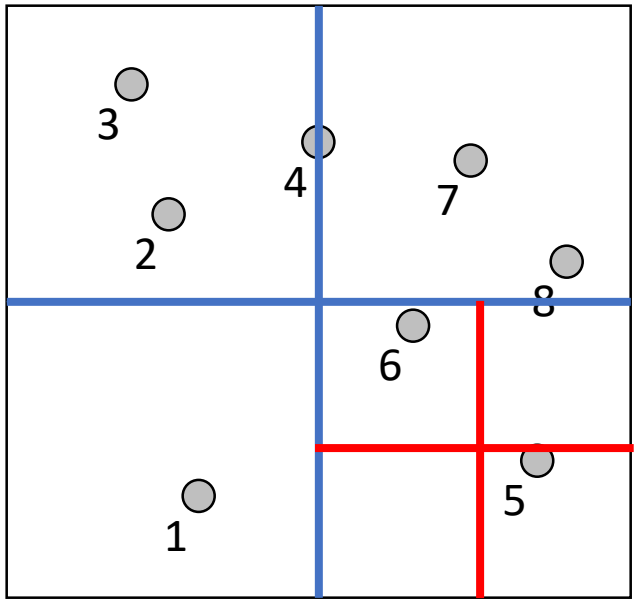


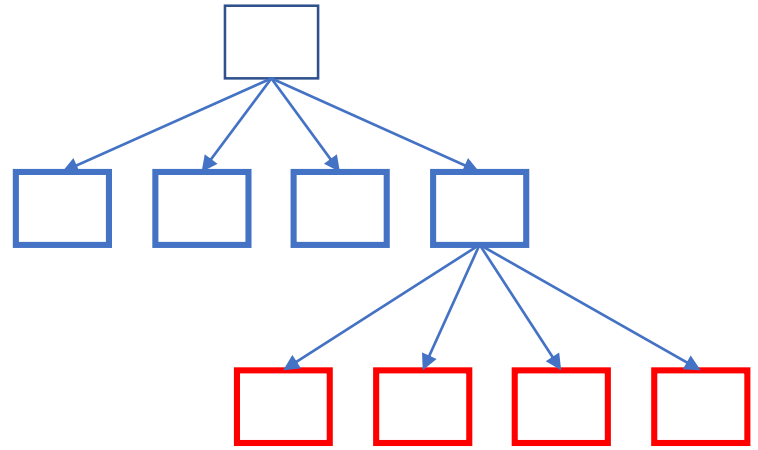
Traditional Index Analogy

- Binary Tree → KD-Tree
- Hash Index

Traditional Index Analogy

- Binary Tree → KD-Tree
- Hash Index → Grid Index
- Binary Tree → Quad Tree
  - Each node divides the space into 4 **quad**rants
  - Each node has 4 children





Traditional Index Analogy

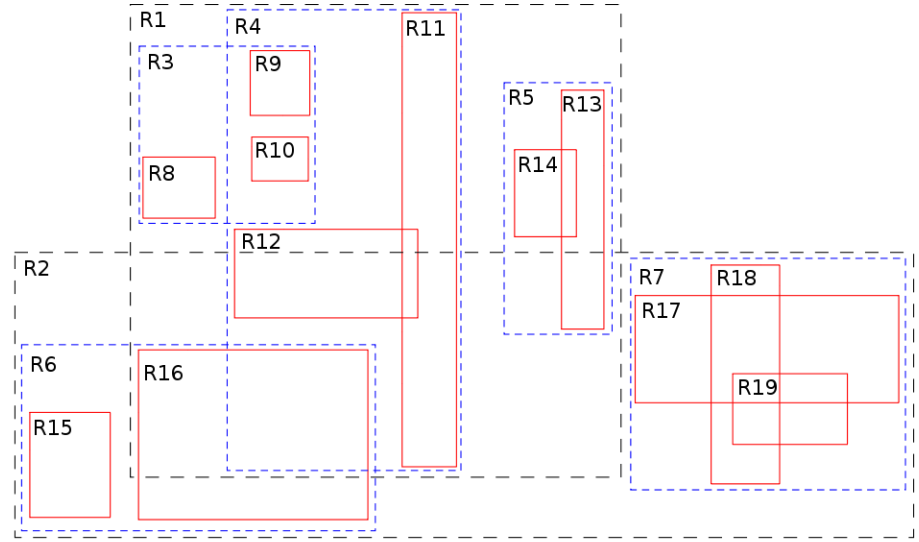
- Binary Tree → KD-Tree
- Hash Index → Grid Index
- Binary Tree → Quad Tree
- B-Tree

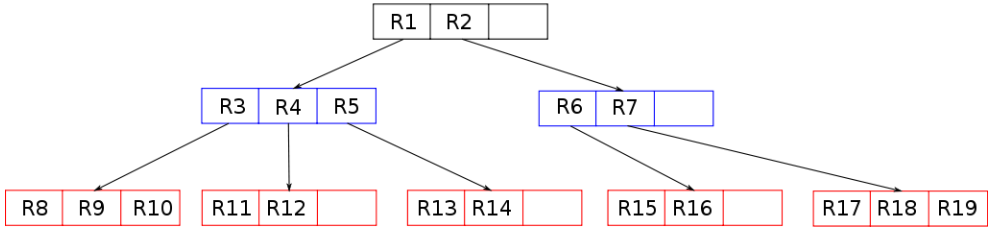
Traditional Index Analogy

- Binary Tree → KD-Tree
- Hash Index → Grid Index
- Binary Tree → Quad Tree
- B-Tree → R-Tree

R-Tree

- Balanced search tree
- The two children of a given node
  - minimizes the bounding box
  - bounding boxes of two nodes can intersect
- Designed for storage on disk
  - Each node corresponds to a “page”
- Every node satisfies a *minimum fill*

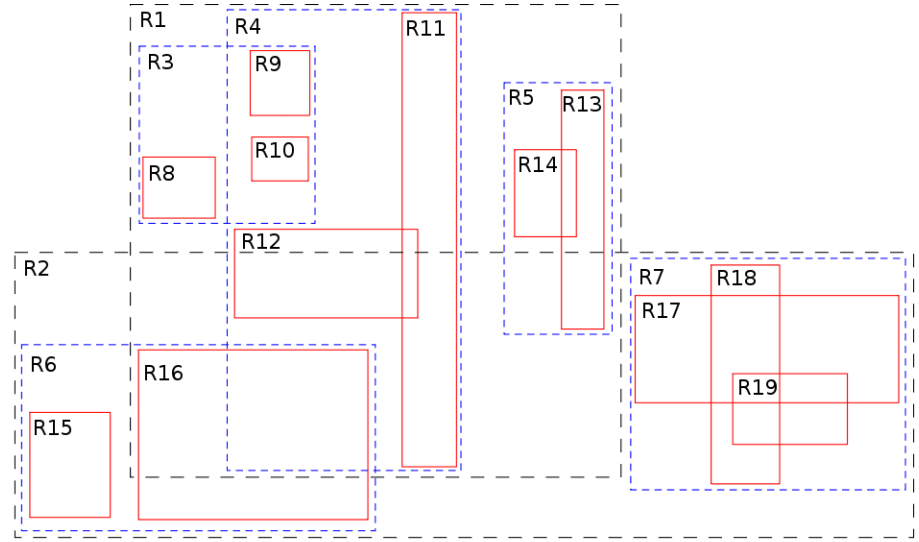


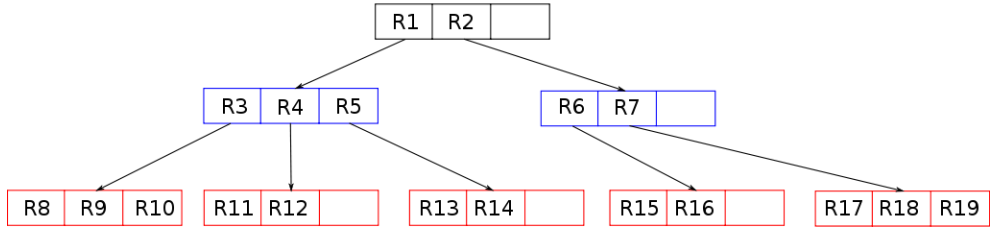


Source: https://en.wikipedia.org/wiki/R-tree

R-Tree

- Advantages
  - Supports addition / removal of data
- Disadvantages
  - Bounding boxes can intersect => queries can get expensive
  - Especially true in higher dimensions



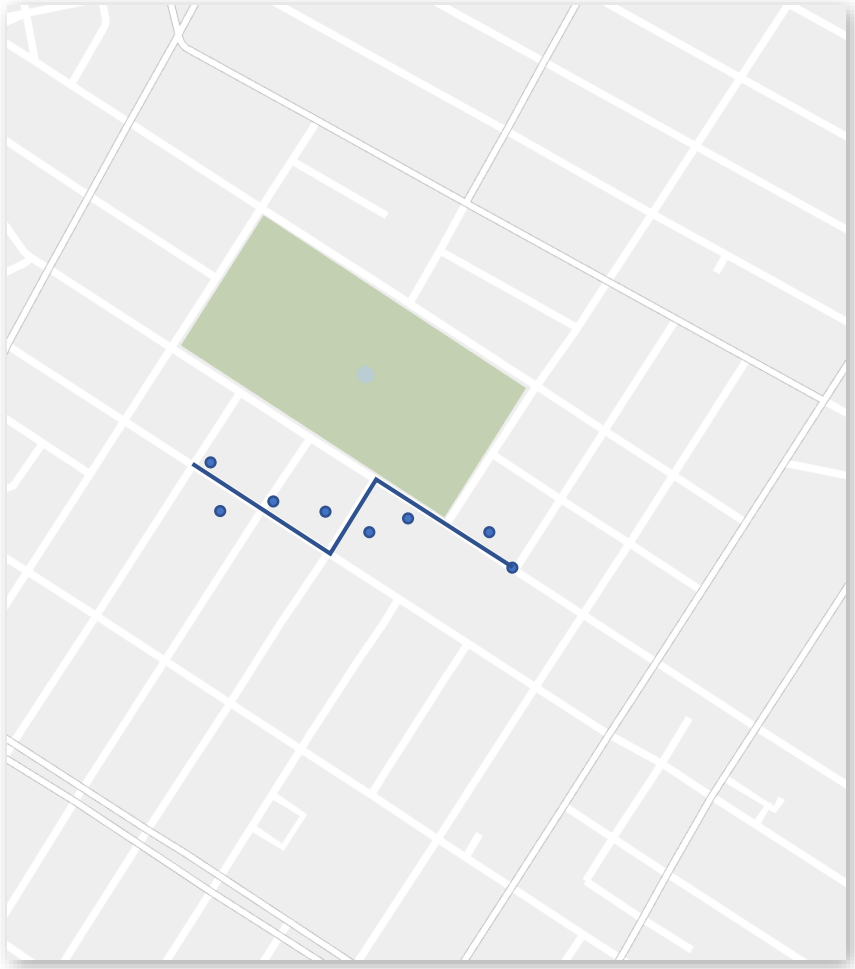


Source: https://en.wikipedia.org/wiki/R-tree

Spatial Data Analysis

Data Cleaning

- Taxi trips
  - Locations fall into the rivers
- Trajectories
  - Map matching
- Polygons
  - Self intersecting polygons

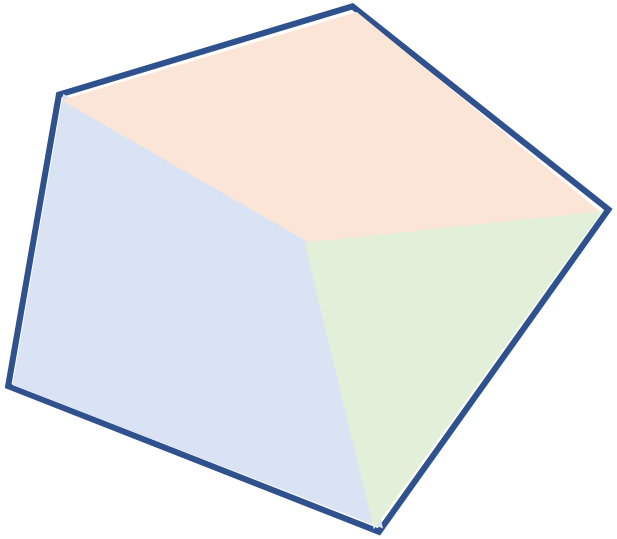


Data Transformations

- Coordinate Reference Systems
  https://spatialreference.org/ref/epsg/
  - Lat, lon - epsg:4326 (WGS84)
  - Mercator - epsg:4326
  - …

Data Transformations

- Different “Resolutions”
- Interpolate
  - Linear
  - Constant



Data Transformations

- Different “Resolutions”
- Interpolate
  - Constant
  - Linear
- Weighted