

Table of Contents

- Spatial Index Recipes 1
 - Using STRtree 1
 - Using Quadtree..... 1
 - Using GeoHash 3

Spatial Index Recipes

Using STRtree

Create a STRtree spatial index

```
STRtree index = new STRtree()
```

Insert Geometries and their Bounds

```
index.insert(new Bounds(0,0,10,10), new Point(5,5))
index.insert(new Bounds(2,2,6,6), new Point(4,4))
index.insert(new Bounds(20,20,60,60), new Point(30,30))
index.insert(new Bounds(22,22,44,44), new Point(32,32))
```

Get the size of the index

```
int size = index.size
println size
```

```
4
```

Query the index

```
List results = index.query(new Bounds(1,1,5,5))
results.each { Geometry geometry ->
    println geometry
}
```

```
POINT (4 4)
POINT (5 5)
```

Using Quadtree

Create a Quadtree spatial index

```
Quadtree index = new Quadtree()
```

Insert Geometries and their Bounds

```
index.insert(new Bounds(0,0,10,10), new Point(5,5))
index.insert(new Bounds(2,2,6,6), new Point(4,4))
index.insert(new Bounds(20,20,60,60), new Point(30,30))
index.insert(new Bounds(22,22,44,44), new Point(32,32))
```

Get the size of the index

```
int size = index.size
println size
```

4

Query the index with a Bounds

```
List results = index.query(new Bounds(1,1,5,5))
results.each { Geometry geometry ->
    println geometry
}
```

```
POINT (30 30)
POINT (32 32)
POINT (5 5)
POINT (4 4)
```

Query the entire index

```
List allResults = index.queryAll()
allResults.each { Geometry geometry ->
    println geometry
}
```

```
POINT (30 30)
POINT (32 32)
POINT (5 5)
POINT (4 4)
```

Remove an item from the index

```
Geometry itemToRemove = allResults[0]
boolean removed = index.remove(itemToRemove.bounds, itemToRemove)
println "Removed? ${removed}"
println "Size = ${index.size}"
```

```
Removed = true  
Size = 3
```

Using GeoHash

Encode a Point as a String

```
GeoHash geohash = new GeoHash()  
Point point = new Point(112.5584, 37.8324)  
String hash = geohash.encode(point)  
println hash
```

```
ww8p1r4t8
```

Decode a String to a Point

```
GeoHash geohash = new GeoHash()  
Point point = geohash.decode("ww8p1r4t8")  
println point
```

```
POINT (112.55838632583618 37.83238649368286)
```

Encode a Point as a Long

```
GeoHash geohash = new GeoHash()  
Point point = new Point(112.5584, 37.8324)  
long hash = geohash.encodeLong(point)  
println hash
```

```
4064984913515641
```

Decode a String to a Point

```
GeoHash geohash = new GeoHash()  
Point point = geohash.decode(4064984913515641)  
println point
```

```
POINT (112.55839973688126 37.83240124583244)
```