

# Table of Contents

- Spatial Index Recipes..... 1
  - Using STRtree ..... 1
  - Using HPRtree..... 1
  - Using Quadtree..... 2
  - Using GeoHash ..... 3

# Spatial Index Recipes

The Index classes are in the [geoscript.index](#) package.

## 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 HPRtree

*Create a HPRtree spatial index*

```
HPRtree index = new HPRtree()
```

### *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 (5 5)
POINT (4 4)
```

## 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 Point from a String*

```
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 Point from a Long*

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

```
POINT (112.55839973688126 37.83240124583244)
```

#### *Decode a Bounds from a String*

```
GeoHash geohash = new GeoHash()  
Bounds bounds = geohash.decodeBounds("ww8p1r4t8")  
println bounds
```

```
(112.55836486816406,37.83236503601074,112.5584077835083,37.83240795135498)
```

#### *Decode a Bounds from a Long*

```
GeoHash geohash = new GeoHash()  
Bounds bounds = geohash.decodeBounds(4064984913515641)  
println bounds
```

```
(112.55836486816406,37.83236503601074,112.5584077835083,37.83240795135498)
```

### *Find neighboring geohash strings*

```
GeoHash geohash = new GeoHash()
String hash = "dqcjg"
String north    = geohash.neighbor(hash, GeoHash.Direction.NORTH)
String northwest = geohash.neighbor(hash, GeoHash.Direction.NORTHWEST)
String west     = geohash.neighbor(hash, GeoHash.Direction.WEST)
String southwest = geohash.neighbor(hash, GeoHash.Direction.SOUTHWEST)
String south    = geohash.neighbor(hash, GeoHash.Direction.SOUTH)
String southeast = geohash.neighbor(hash, GeoHash.Direction.SOUTHEAST)
String east     = geohash.neighbor(hash, GeoHash.Direction.EAST)
String northeast = geohash.neighbor(hash, GeoHash.Direction.NORTHEAST)
String str = ""
    | ${northwest} ${north} ${northeast}
    | ${west} ${hash} ${east}
    | ${southwest} ${south} ${southeast}
    |"".stripMargin()
println str
```

```
dqcjt dqcjw dqcjx
dqejm dqcjg dqcjr
dqcyj dqcjm dqcjp
```

### *Find neighboring geohash longs*

```
GeoHash geohash = new GeoHash()
Long hash = 1702789509
Long north    = geohash.neighbor(hash, GeoHash.Direction.NORTH)
Long northwest = geohash.neighbor(hash, GeoHash.Direction.NORTHWEST)
Long west     = geohash.neighbor(hash, GeoHash.Direction.WEST)
Long southwest = geohash.neighbor(hash, GeoHash.Direction.SOUTHWEST)
Long south    = geohash.neighbor(hash, GeoHash.Direction.SOUTH)
Long southeast = geohash.neighbor(hash, GeoHash.Direction.SOUTHEAST)
Long east     = geohash.neighbor(hash, GeoHash.Direction.EAST)
Long northeast = geohash.neighbor(hash, GeoHash.Direction.NORTHEAST)
String str = ""
    | ${northwest} ${north} ${northeast}
    | ${west} ${hash} ${east}
    | ${southwest} ${south} ${southeast}
    |"".stripMargin()
println str
```

```
1702789434 1702789520 1702789522
1702789423 1702789509 1702789511
1702789422 1702789508 1702789510
```

*Find all neighboring geohash strings*

```
GeoHash geohash = new GeoHash()
String hash = "dqcjq"
Map neighbors = geohash.neighbors(hash)
String north    = neighbors[GeoHash.Direction.NORTH]
String northwest = neighbors[GeoHash.Direction.NORTHWEST]
String west     = neighbors[GeoHash.Direction.WEST]
String southwest = neighbors[GeoHash.Direction.SOUTHWEST]
String south    = neighbors[GeoHash.Direction.SOUTH]
String southeast = neighbors[GeoHash.Direction.SOUTHEAST]
String east     = neighbors[GeoHash.Direction.EAST]
String northeast = neighbors[GeoHash.Direction.NORTHEAST]
String str = ""
            | ${northwest} ${north} ${northeast}
            | ${west}    ${hash} ${east}
            | ${southwest} ${south} ${southeast}
            |"".stripMargin()
println str
```

```
dqcjt dqcjw dqcjx
dqejm dqcjq dqcjr
dqcjj dqcjn dqcjp
```

*Find all neighboring geohash longs*

```
GeoHash geohash = new GeoHash()
Long hash = 1702789509
Map neighbors = geohash.neighbors(hash)
Long north    = neighbors[GeoHash.Direction.NORTH]
Long northwest = neighbors[GeoHash.Direction.NORTHWEST]
Long west     = neighbors[GeoHash.Direction.WEST]
Long southwest = neighbors[GeoHash.Direction.SOUTHWEST]
Long south    = neighbors[GeoHash.Direction.SOUTH]
Long southeast = neighbors[GeoHash.Direction.SOUTHEAST]
Long east     = neighbors[GeoHash.Direction.EAST]
Long northeast = neighbors[GeoHash.Direction.NORTHEAST]
String str = ""
            | ${northwest} ${north} ${northeast}
            | ${west}    ${hash} ${east}
            | ${southwest} ${south} ${southeast}
            |"".stripMargin()
println str
```

```
1702789434 1702789520 1702789522
1702789423 1702789509 1702789511
1702789422 1702789508 1702789510
```

*Find all geohashes as strings within a Bounds*

```
GeoHash geohash = new GeoHash()
List<String> bboxes = geohash.bboxes(new Bounds(120, 30, 120.0001, 30.0001), 8)
bboxes.each { String hash ->
    println hash
}
```

```
wtm6dtm6
wtm6dtm7
```

*Find all geohashes as longs within a Bounds*

```
GeoHash geohash = new GeoHash()
List<Long> bboxes = geohash.bboxesLong(new Bounds(120, 30, 120.0001, 30.0001), 40)
bboxes.each { long hash ->
    println hash
}
```

```
989560464998
989560464999
```