



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

ESCUELA PROFESIONAL DE CIENCIA DE LA
COMPUTACIÓN

ESTRUCTURA DE DATOS AVANZADAS

KD TREE

Alumnos:

Angelo Aldo Perez Rodriguez

Josnick Chayña Batallanes

Rodrigo Alexander Mamani Sucacahua

Shirley Michelle Oxa Cacya

Renato Gonzalo Cespedes Fuentes

Octubre 2020

Contents

	1
1 Introducción	2
2 Implemente la función range query circle	2
3 Implemente la función range query rectangle	3
4 Resultados	4
5 Repositorio	5

Figure 1: Caption

1 Introducción

La función RangeQuery , devuelve un conjunto de puntos a partir de una área de búsqueda, por ejemplo en la Figura de abajo, el círculo y el rectángulo representan el rango de búsqueda. La estructura debe ser capaz de retornar el conjunto de puntos dentro de dicho rango sin consultar todos los puntos.

2 Implemente la función range query circle

```
1  function range_query_circle(node, point, kpoints , radio, depth =
    0)
2  {
3      if(node==null)
4          return null;
5
6      var next_branch;
7      var opposite_branch;
8      var temp;
9
10     if(point[depth % k]< node.point[depth % k])
11     { next_branch = node.left;
12       opposite_branch = node.rigth;
13     }
14
15     else
16     {
17         next_branch = node.rigth;
18         opposite_branch = node.left;
19     }
20
21
22     closest_range(point, range_query_circle(next_branch, point,
        kpoints, radio, depth +1), node.point);
23     //count++;
24
25     if(distanceSquared(point,node.point)<radio)
26     {
27         node.point.push(distanceSquared(point,node.point));
28         kpoints.push(node.point);
29     }
30
31     if(radio>=Math.abs(point[depth%k]-node.point[depth%k]))
32     {
33         closest_range(point, range_query_circle(opposite_branch, point,
            kpoints, radio, depth +1), node.point);
34     }
35 }
36
37
38 }
```

3 Implemente la función range query rectangle

```
1 function range_query_rectangle(node, point, kpoints, rectangle,
   depth = 0)
2 {
3   if(node==null)
4     return null;
5
6   var next_branch;
7   var opposite_branch;
8   var temp;
9
10  if(point[depth % k] < node.point[depth % k])
11  { next_branch = node.left;
12    opposite_branch = node.rigth;
13  }else{
14    next_branch = node.rigth;
15    opposite_branch = node.left;
16  }
17  closest_range(point, range_query_rectangle(next_branch, point,
18    kpoints, rectangle, depth +1), node.point);
19
20  if(node.point[0] > (rectangle.x-rectangle.w) && node.point[0] < (
21    rectangle.x+rectangle.w) && node.point[1] > (
22    rectangle.y-rectangle.h) && node.point[1] < (rectangle.y+
23    rectangle.h))
24  {
25    kpoints.push(node.point);
26  }
27
28  if(rectangle.x+rectangle.w >= Math.abs(point[depth%k]-node.point[
29    depth%k]) || rectangle.x-rectangle.w >= Math.abs(point[depth%k]
30    -node.point[depth%k]) || rectangle.y+rectangle.h >= Math.abs(
31    point[depth%k]-node.point[depth%k]) ||
32    rectangle.y-rectangle.h >= Math.abs(point[depth%k]-node.point[
33    depth%k]))
34  {
35    closest_range(point, range_query_rectangle(opposite_branch,
36      point, kpoints, rectangle, depth +1), node.point);
37  }
38 }
```

4 Resultados

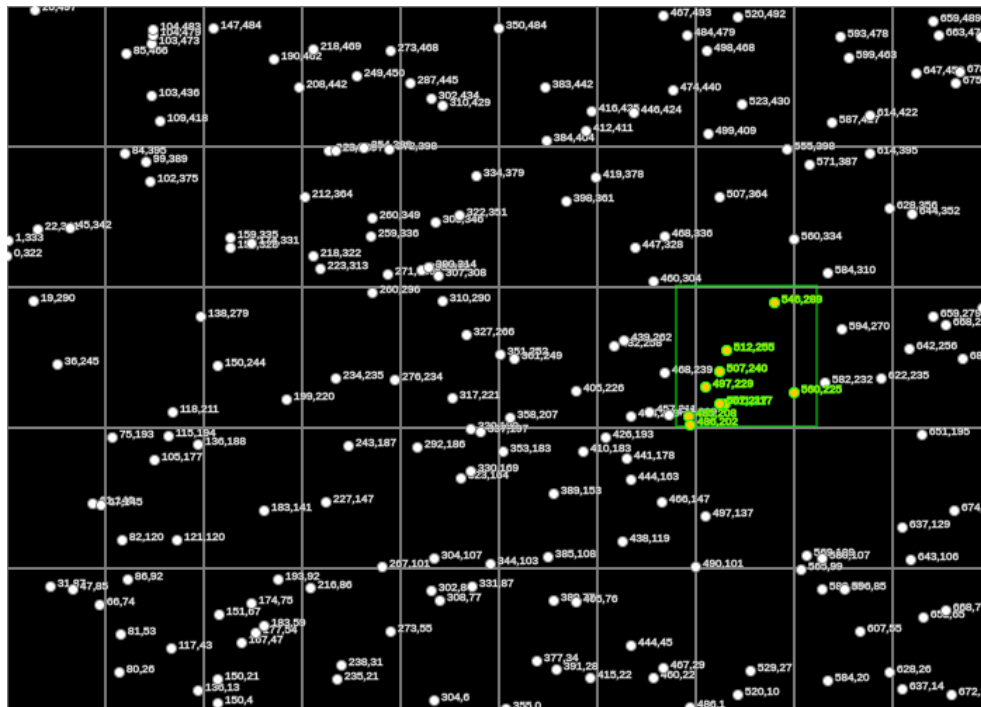


Figure 2: Los vecinos mas cercanos en un cierto rango

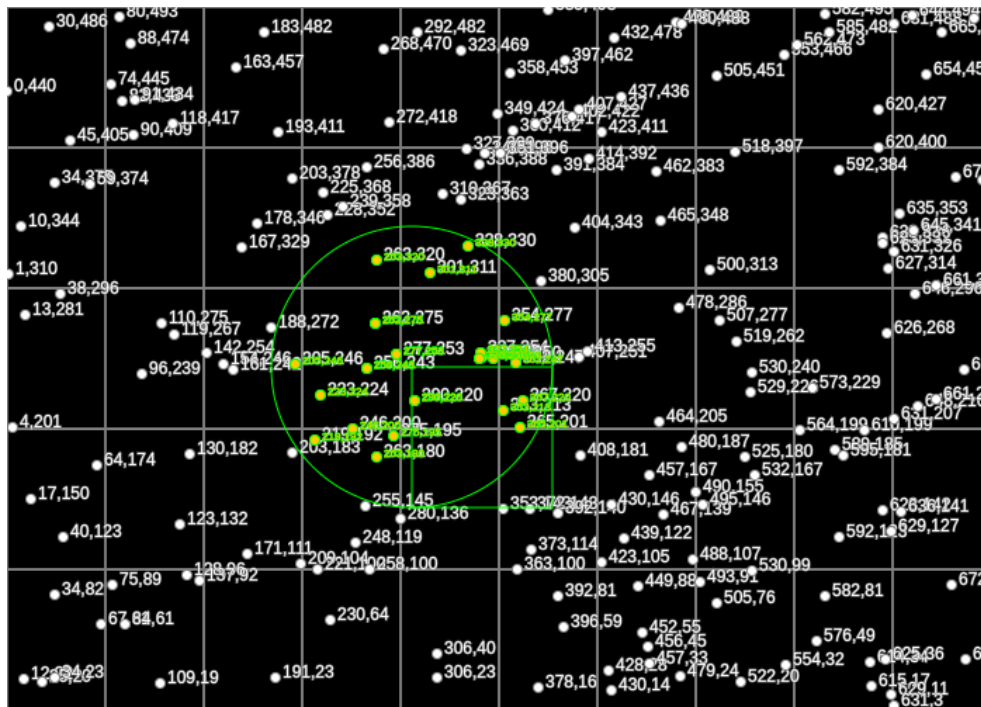


Figure 3: Obtención de Kn puntos mas cercanos donde kn=20, mostrando desde consola sus coordenadas y la distancia con respecto al punto 140,90

5 Repositorio

- **Practica 7:**

El repositorio donde la practica 7 esta, es en el siguiente [link](#)