

UNIVERSIDAD NACIONAL DE SAN AGUSTIN DE
AREQUIPA



ESTRUCTURA DE DATOS AVANZADOS

QuadTree

Alumna :

Chullunquía Rosas, Sharon Rossely

Profesor :

Machaca Arceda, Vicente Enrique

4 de octubre de 2020

Pregunta #01

Edite el archivo quadtree.js y complete la función query.

Respuesta:

```
1 query(range, found) {
2   if (!found) {
3     found = [];
4   }
5
6   if (!range.intersects(this.boundary)) {
7     return found;
8   }
9
10  for (let p of this.points) {
11    if (range.contains(p)) {
12      found.push(p);
13    }
14  }
15
16  if (this.divided) {
17    this.northwest.query(range, found);
18    this.northeast.query(range, found);
19    this.southwest.query(range, found);
20    this.southeast.query(range, found);
21  }
22  return found;
23 }
```

Pregunta #02

Edite el archivo sketch.js con el siguiente código. Muestre sus resultados y comente.

Respuesta:

```
1 let qt;
2 let count = 0;
3
4 function setup() {
5   createCanvas(400, 400);
6
7   // centre point and half of width and height
8   let boundary = new Rectangle(200, 200, 200, 200);
9
10  // each leave just could have 4 elements
11  qt = new QuadTree(boundary, 4);
12
13  console.log(qt);
14 }
```

```
15   for (let i = 0; i < 50; i++) {
16     let p = new Point(Math.random() * 400, Math.random() * 400);
17     qt.insert(p);
18   }
19
20 }
21
22 function draw() {
23   background(0);
24   qt.show();
25
26   stroke(0, 255, 0);
27   rectMode(CENTER);
28   let range = new Rectangle(random(200), random(200), random(50)
29     , random(50))
29   rect(range.x, range.y, range.w * 2, range.h * 2);
30   let points = [];
31   qt.query(range, points);
32   for (let p of points) {
33     strokeWeight(4);
34     point(p.x, p.y);
35   }
36
37   console.log(count);
38 }
```

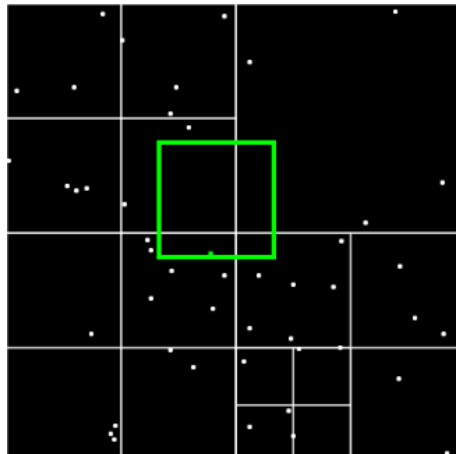


Figura 1: Visualización del QuadTree.

Pregunta #03

En este caso vamos a verificar cuantas veces se consulta un punto en la función query, para esto usaremos la variable global **count** (ya esta definida en sketch.js). Agregue una instrucción en la función query donde se incremente el valor de **count**.

Luego evalúe y verifique cuantas veces se consultada un punto, prueba con mas de 1000 puntos en el Quadtree.

Respuesta:

```
1 query(range, found) {
2     if (!found) {
3         found = [];
4     }
5
6     if (!range.intersects(this.boundary)) {
7         return found;
8     }
9
10    for (let p of this.points) {
11        if (range.contains(p)) {
12            found.push(p);
13            count++; // Contando cuantas veces se consulta
                     // un punto
14        }
15    }
16
17    if (this.divided) {
18        this.northwest.query(range, found);
19        this.northeast.query(range, found);
20        this.southwest.query(range, found);
21        this.southeast.query(range, found);
22    }
23    return found;
24 }
```

```
1 let qt;
2 let count = 0;
3
4 function setup() {
5     createCanvas(400, 400);
6
7     // centre point and half of width and height
8     let boundary = new Rectangle(200, 200, 200, 200);
9
10    // each leave just could have 4 elements
11    qt = new QuadTree(boundary, 4);
12
13    console.log(qt);
14
15    for (let i = 0; i < 1100; i++) { // Mas de 1000 puntos
16        let p = new Point(Math.random() * 400, Math.random() * 400);
17        qt.insert(p);
18    }
19
20 }
21
```

```
22 function draw() {  
23   background(0);  
24   qt.show();  
25  
26   stroke(0, 255, 0);  
27   rectMode(CENTER);  
28   let range = new Rectangle(random(200), random(200), random(50)  
29     , random(50))  
29   rect(range.x, range.y, range.w * 2, range.h * 2);  
30   let points = [];  
31   qt.query(range, points);  
32   for (let p of points) {  
33     strokeWeight(4);  
34     point(p.x, p.y);  
35   }  
36  
37   console.log(count);  
38 }
```

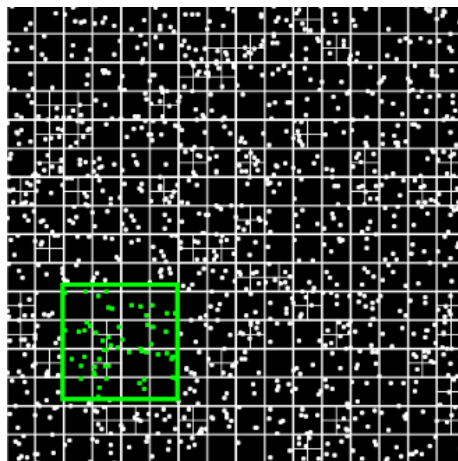


Figura 2: Visualización del QuadTree con mas de 1000 puntos.

Pregunta #04

Editamos el archivo sketch.js, En este caso haremos consultas con el mouse. Muestre sus resultados y comente.

Respuesta:

```
1 let qt;  
2 let count = 0;  
3  
4 function setup() {  
5   createCanvas(400, 400);  
6 }
```

```
7 // centre point and half of width and height
8 let boundary = new Rectangle(200, 200, 200, 200);
9
10 // each leave just could have 4 elements
11 qt = new QuadTree(boundary, 4);
12
13 console.log(qt);
14
15 for (let i = 0; i < 1100; i++) { // Mas de 1000 puntos
16     let p = new Point(Math.random() * 400, Math.random() * 400);
17     qt.insert(p);
18 }
19
20 }
21
22 function draw() {
23     background(0);
24     qt.show();
25
26     stroke(0, 255, 0);
27     rectMode(CENTER);
28     let range = new Rectangle(mouseX, mouseY, 50, 50)
29     rect(range.x, range.y, range.w * 2, range.h * 2);
30     let points = [];
31     qt.query(range, points);
32     for (let p of points) {
33         strokeWeight(4);
34         point(p.x, p.y);
35     }
36
37     console.log(count);
38 }
```

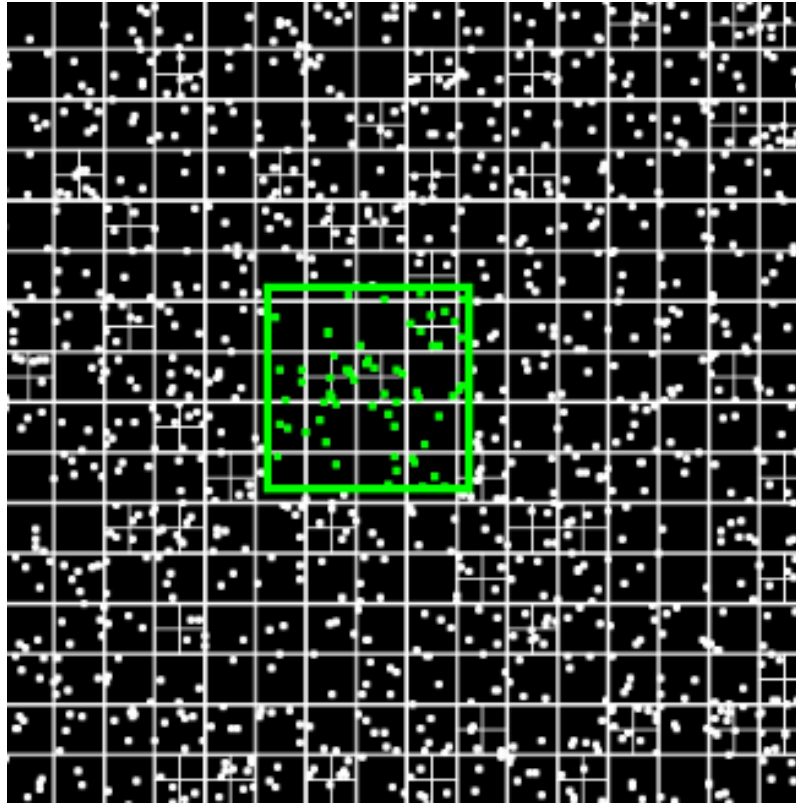


Figura 3: Consultas con el mouse.

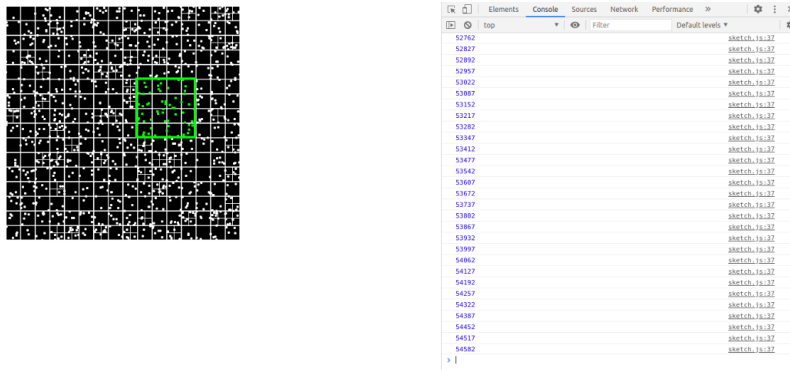


Figura 4: Resultado en consola (count).

Referencias

- [1] Repositorio de la Práctica Nro. 02 en GitHub: <https://github.com/sharon1160/Advanced-Data-Structure/tree/master/Practices/Practice%2003>