

“AÑO DEL BICENTENARIO DEL PERÚ: 200 AÑOS DE INDEPENDENCIA”.



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE
AREQUIPA**

**ESCUELA PROFESIONAL DE CIENCIA DE LA
COMPUTACIÓN
ESTRUCTURA DE DATOS AVANZADAS**

QuadTree

Alumnos:

Angel Andres Bejar Merma

Docente:

Dra. Evelyn Perez C.

28 de octubre de 2021



Índice

1. Marco teórico	2
2. Metodología	4
3. Resultados	5
3.1. probando con 50 elementos	5
4. discusión	5
5. conclusión	5

1. Marco teórico

Un Quadtree es una estructura de datos en la que cada nodo tiene exactamente cuatro hijos ver figura 1.

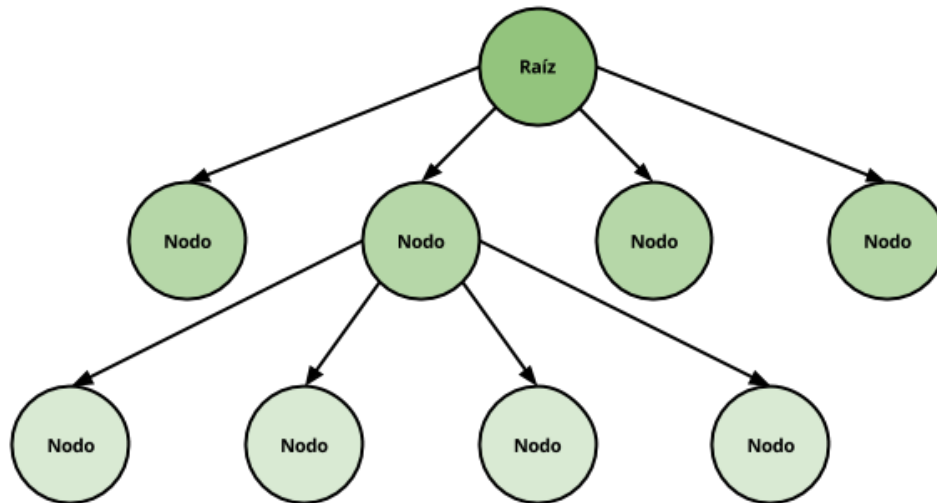


Figura 1: arbol quadtree

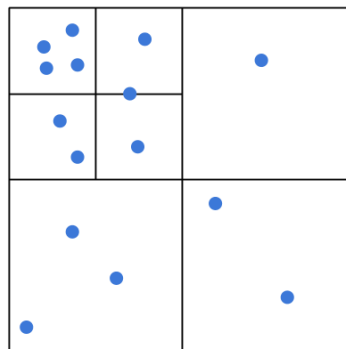


Figura 2: representacion del mapa

Se construye dividiendo continuamente cada nodo hasta que cada nodo hoja solo tenga un único nodo dentro de él. Los Quadtrees se generalizan como árboles “kd / k-dimensionales” cuando tienes más de 4 divisiones en cada nodo;

Aunque generalmente se usan en dos dimensiones, los quadrees se pueden emplear a una cantidad arbitraria de dimensiones. Lo mejor de los Quadtree es la reducción dimensional". En lugar de operar en $O(n^2)$ para una búsqueda lineal común en dos dimensiones, el Quadtree puede lograr un tiempo cercano a $O(\log n)$ para la mayoría de las operaciones.

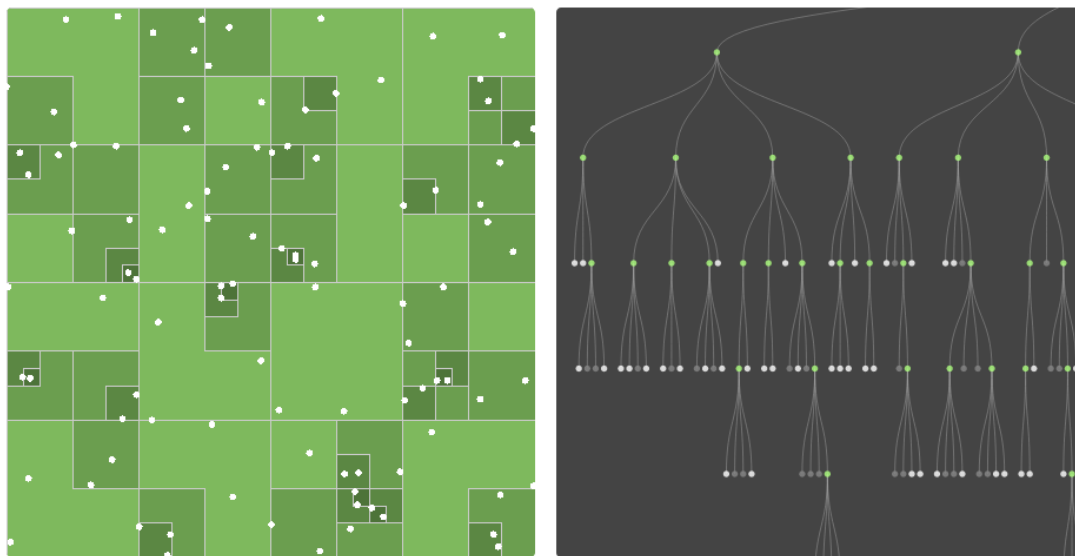


Figura 3: mejor representacion del quadtree

En la figura 3 cada nodo verde representa un rectángulo en un espacio. Cada nodo blanco representa un punto en ese mismo espacio. Los nodos verdes pueden tener hasta cuatro hijos, ya sean blancos o verdes. Estos hijos están contenidos dentro del rectángulo que representa el nodo verde. (Los nodos grises representan espacios vacíos, lugares para nodos hijos que aún no están ocupados).

Todas las formas de quadtree comparten las siguientes características[5] :

- Descomponen el espacio en celdas adaptables.
- Cada celda (o cuadrante) tiene una capacidad máxima. Cuando se alcanza la capacidad máxima, el cubo se divide
- El directorio del árbol sigue la descomposición espacial del quadtree.

Algunos posibles usos de eso incluyen:

- **Detección de colisiones:** Normalmente para detectar una colisión, se compararía todos los puntos contra todos y si fueran 1000 puntos y ninguno de ellos fuera una colisión se tendría que hacerse 1000×1000 comprobaciones lo que es un desperdicio de tiempo y rendimiento.

Así que aquí entra en acción el Quadtree, cuando tenga que buscar, dividirá el mapa en cuadrantes y así hará una comprobación por cuadrante ahorrando tiempo y mejorando el costo. Luego, descubrirá en qué cuadrante dentro de ese cuadrante se encuentra y así recursivamente hasta llegar a la colisión.

En ese sentido solo se tendrá que hacer esto como máximo siete veces para un espacio de 100×100 (asumiendo que los puntos solo pueden tener valores enteros), incluso si hay 1000 puntos en él. Para un espacio de $1,000,000 \times 1,000,000$, es un máximo de 20 veces.

- **Encontrar al vecino más cercano:** Se tiene un mapa de puntos, en lugar de preguntarse si alguno de ellos coincide con un punto dado, podrías pensar se cuál es el punto más cercano que tienes a un punto arbitrario entre tus puntos.

2. Metodología

Para una mejor comprensión del funcionamiento del quadtree puede visitar [4]. En la metodología :

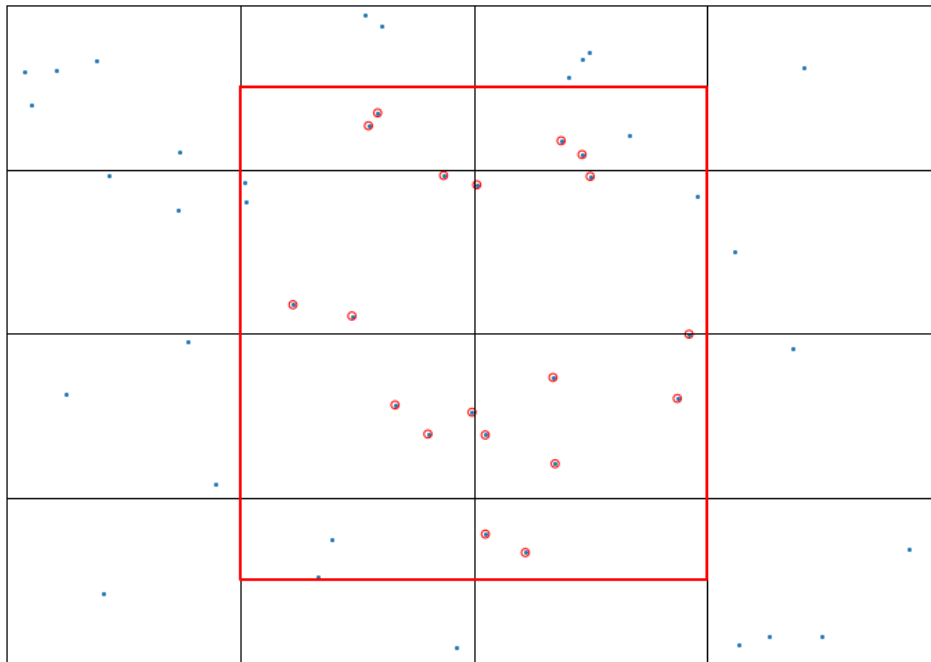
- Se probó empleo Python en una plataforma linux y creando un entorno virtual [2]
- La computadora que utilizo fue una intel i5 de quinta generación con 2 cores ,8 gb de memoria ram.
- Se creó dos archivos el primero para probar, test.py donde se importan las funciones definidas en las clases del segundo archivo quadtree.py.
- *La estructura de código es el siguiente* : Se implementó tres principales clases: punto ,rectángulo y Quadtree [3]. En esta última se definieron las funciones más importantes *insertar* y *consultar rango*. Para la primera parte se crearon puntos de forma aleatoria,

Para la **inserción**, primero se verifica si el nodo dado está dentro de los límites del quadtree actual . Si no es así, dejamos de insertar inmediatamente y si sobre pasa la capacidad llamamos a la función divide para dividir la celda y empezar de nuevo el proceso. Entonces si está dentro de los límites, seleccionamos el hijo apropiado para contener este nodo en función de su ubicación. Esta función es $O(\log N)$ donde N es el tamaño de la distancia.

3. Resultados

3.1. probando con 50 elementos

Estableciendo $N=50$.



```
(envPython) angel@debianita:~/enviarEstructuradeDatosJue16/quadTree$ python test.py
Total puntos: 50
puntos en rango: 13
```

En la figura el rango o sea la cantidad de puntos dentro del cuadrado rojo son 13 [1].

4. discusión

Quadtree no trabaja bien cuando debe almacenar gran cantidad de diferentes píxeles existentes en una imagen con muchos colores, dado que podría tomar un tamaño excesivamente grande. También por ser una estructura de datos de memoria principal, al querer representar imágenes muy grandes, muchas veces el Quadtree no puede ser almacenado en dicha memoria. En esos casos, es mejor usar estructuras alternativas por ejemplo, un B+ tree, para compensar esa limitación.

5. conclusión

Los Quadrees son una forma de dividir el espacio para que sea fácil de intersecar y buscar. La detección de colisiones puede ser una operación costosa y puede ralentizar el rendimiento. Los Quadrees son una forma en que puede ayudar a acelerar la detección de colisiones y mantener por ejemplo un juego funcionando a la máxima velocidad.

Referencias

- [1] GeminiCCCC. Quadtree . <https://gist.github.com/GeminiCCCC>. [fuente original].
- [2] happy coding. video. https://www.youtube.com/watch?v=9CJp-k_OjC4. [Online; accessed 27-setiembre-2021].
- [3] jrtechs. Implementing a Quadtree in Python. <https://jrtechs.net/data-science/implementing-a-quadtree-in-python#:~:text=A%20quadtree%20is%20a%20data,single%20unit%20of%20spatial%20information>. [Online; accessed 27-setiembre-2021].
- [4] Opensa. visualizacion Quadtree ,The PR Quadtree . <https://opensa-server.cs.vt.edu/ODSA/Books/CS3/html/PRquadtree.html>. [Online; accessed 27-setiembre-2021].
- [5] Wikipedia. Quadtree . <https://es.wikipedia.org/wiki/Quadtree>. [Online; accessed 27-setiembre-2021].