

## Práctica 04

DOCENTE	CARRERA	CURSO
Vicente Machaca Arceda	Maestría en Ciencias de la Computación	Algoritmos y Estructura de Datos

PRÁCTICA	TEMA	DURACIÓN
04	Kd-tree	—

### 1. Integrantes

- Grupo N° 2
- Integrantes:
  - EDER ALONSO AMPUERO ATAMARI
  - HOWARD FERNANDO ARANZAMENDI MORALES
  - JOSE EDISON PEREZ MAMANI
  - HENRRY IVAN ARIAS MAMANI

### 2. Repositorio GitHub

URL Github: Repositorio Práctica 4 AyED

### 3. Marco Teórico

#### 3.1. Kd-Tree

##### 3.1.1. Definición

Un KD-Tree (también llamado árbol K-dimensional) es un árbol de búsqueda binaria donde los datos en cada nodo son un punto K-dimensional en el espacio. En resumen, es una estructura de datos de partición de espacio (detalles a continuación) para organizar puntos en un espacio K-Dimensional.

Un nodo que no es una hoja en el árbol K-D divide el espacio en dos partes, llamadas medios espacios.

Los puntos a la izquierda de este espacio están representados por el subárbol izquierdo de ese nodo y los puntos a la derecha del espacio están representados por el subárbol derecho. Pronto estaremos explicando el concepto de cómo se divide el espacio y se forma el árbol.

En aras de la simplicidad, entendamos un árbol 2-D con un ejemplo.

La raíz tendría un plano alineado con el eje x, los hijos de la raíz tendrían ambos planos alineados con el eje y, los nietos de la raíz tendrían todos planos alineados con el eje x, y los bisnietos de la raíz tendrían todos planos alineados con el eje y, y así sucesivamente.

### 3.1.2. Generalización

Numeremos los planos como 0, 1, 2, ... (K - 1). Del ejemplo anterior, es bastante claro que un punto (nodo) en la profundidad D tendrá un plano alineado donde A se calcula como:

$$A = D \bmod K$$

### 3.1.3. ¿Cómo determinar si un punto estará en el subárbol izquierdo o en el subárbol derecho?

Si el nodo raíz está alineado en el plano A, el subárbol izquierdo contendrá todos los puntos cuyas coordenadas en ese plano sean más pequeñas que las del nodo raíz. De manera similar, el subárbol derecho contendrá todos los puntos cuyas coordenadas en ese plano sean mayores-iguales que las del nodo raíz.

### 3.1.4. Creación de un árbol 2-D

Considere los siguientes puntos en un plano 2-D: (3, 6), (17, 15), (13, 15), (6, 12), (9, 1), (2, 7), (10, 19)

1. Insertar (3, 6): dado que el árbol está vacío, conviértalo en el nodo raíz.
2. Insertar (17, 15): compararlo con el punto del nodo raíz. Dado que el nodo raíz está alineado con X, el valor de la coordenada X se comparará para determinar si se encuentra en el subárbol derecho o en el subárbol izquierdo. Este punto estará alineado con Y.
3. Insertar (13, 15): el valor X de este punto es mayor que el valor X del punto en el nodo raíz. Entonces, esto estará en el subárbol derecho de (3, 6). Nuevamente compare el valor Y de este punto con el valor Y del punto (17, 15) (¿Por qué?). Como son iguales, este punto estará en el subárbol derecho de (17, 15). Este punto estará alineado con X.
4. Insertar (6, 12): el valor X de este punto es mayor que el valor X del punto en el nodo raíz. Entonces, esto estará en el subárbol derecho de (3, 6). Nuevamente compare el valor Y de este punto con el valor Y del punto (17, 15) (¿Por qué?). Como 12 < 15, este punto estará en el subárbol izquierdo de (17, 15). Este punto estará alineado con X.
5. Insertar (9, 1): De manera similar, este punto estará a la derecha de (6, 12).
6. Insertar (2, 7): De manera similar, este punto estará a la izquierda de (3, 6).
7. Inserta (10, 19): De manera similar, este punto estará a la izquierda de (13, 15).

### 3.1.5. ¿Cómo se divide el espacio?

Los 7 puntos se trazarán en el plano X-Y de la siguiente manera:

1. El punto (3, 6) dividirá el espacio en dos partes: Dibujar la línea  $X = 3$ .
2. El punto (2, 7) dividirá el espacio a la izquierda de la línea  $X = 3$  en dos partes horizontalmente. Dibuja la línea  $Y = 7$  a la izquierda de la línea  $X = 3$ .
3. El punto (17, 15) dividirá el espacio a la derecha de la línea  $X = 3$  en dos partes horizontalmente. Dibuja la línea  $Y = 15$  a la derecha de la línea  $X = 3$ .
4. El punto (6, 12) dividirá el espacio debajo de la línea  $Y = 15$  ya la derecha de la línea  $X = 3$  en dos partes. Dibuja la línea  $X = 6$  a la derecha de la línea  $X = 3$  y debajo de la línea  $Y = 15$ .



Figura 1: Nodos en KD-TREE

5. El punto  $(13, 15)$  dividirá el espacio debajo de la línea  $Y = 15$  ya la derecha de la línea  $X = 6$  en dos partes. Dibuja la línea  $X = 13$  a la derecha de la línea  $X = 6$  y debajo de la línea  $Y = 15$ .
6. El punto  $(9, 1)$  dividirá el espacio entre las líneas  $X = 3$ ,  $X = 6$  e  $Y = 15$  en dos partes. Dibuja la línea  $Y = 1$  entre las líneas  $X = 3$  y  $X = 13$ .
7. El punto  $(10, 19)$  dividirá el espacio a la derecha de la línea  $X = 3$  y arriba de la línea  $Y = 15$  en dos partes. Dibuja la línea  $Y = 19$  a la derecha de la línea  $X = 3$  y arriba de la línea  $Y = 15$ .

### 3.2. KNN

#### Definición

El algoritmo de k-vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro.

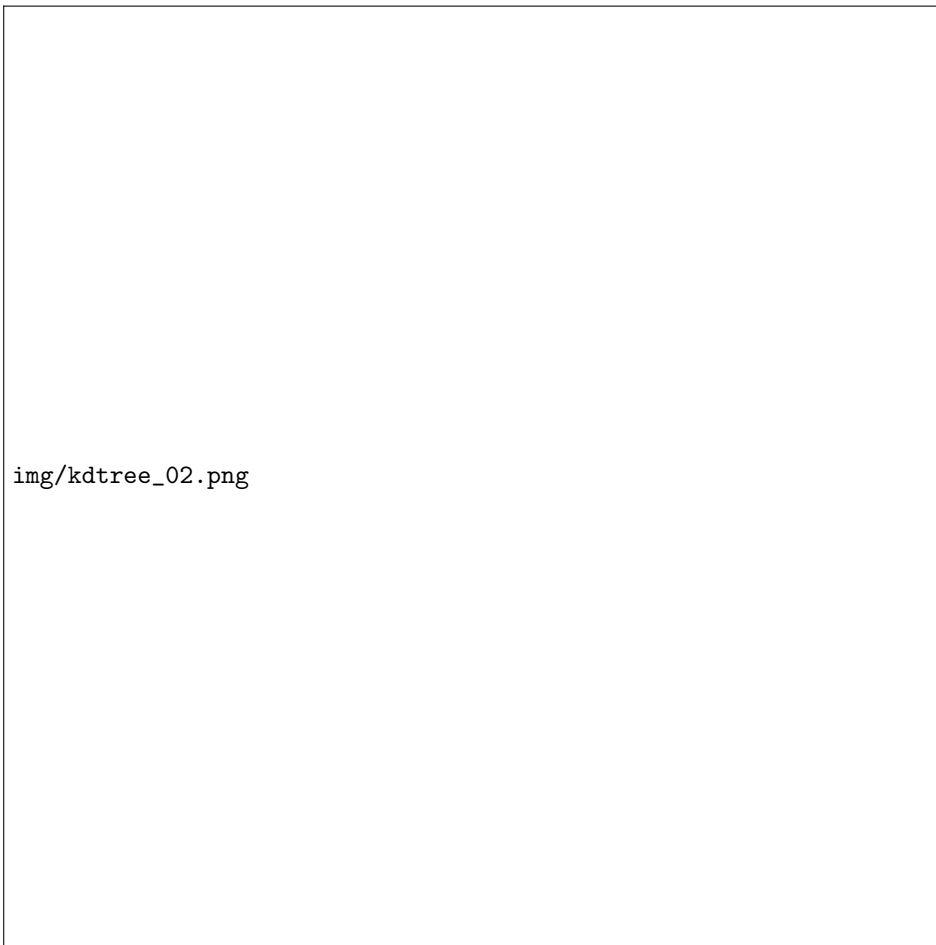


Figura 2: División del espacio en KD-TREE

## 4. Ejercicios

- 4.1. Ejercicio N° 1
- 4.2. Ejercicio N° 2
- 4.3. Ejercicio N° 3
- 4.4. Ejercicio N° 4
- 4.5. Ejercicio N° 5
- 4.6. Ejercicio N° 6
- 4.7. Ejercicio N° 7
- 4.8. Ejercicio N° 8
- 4.9. Ejercicio N° 9
- 4.10. Ejercicio N° 10

## 5. Conclusiones

- El quadtree se codifican en un árbol de cuatro hijos no balanceado, y el octree se codifica en un árbol octal no balanceado.
- De la bibliografía revisada, se concluye que las aplicaciones más comunes para los QuadTree y OcTree son el Procesamiento de imágenes, generación de mallas e indexado espacial.
- Las estructuras octree son usadas mayormente para partir un espacio tridimensional, dividiéndolo recursivamente en ocho octantes, siendo análogas tridimensionales de los quadtree bidimensionales.

## 6. Referencias

1. Tobler, W., Chen, Z. T. (1986). A quadtree for global information storage. *Geographical Analysis*, 18(4), 360-371.
2. Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2), 187-260.
3. Rojas Delgado, J. (2015). Estructura de datos espacial jerárquica para la indexación de agrupaciones de objetos geométricos (Bachelor's thesis, Universidad de las Ciencias Informáticas. Facultad 6).
4. Castillo Oliva, A. D. (2018). Visualización de grandes volúmenes de datos empleando Octree (Bachelor's thesis, Universidad de las Ciencias Informáticas. Centro Vertex, Entornos Interactivos 3D. Facultad 4).
5. Rivero, J. P. S., de la Hoz, Á. P., Medina, M. Á. P. GENERACIÓN DE MALLAS APLICACIONES A LA INGENIERÍA.
6. Brunet Crosa, P., Santistevé i Puyuelo, F., Vilanova, A., Chiarabini, L., Patow, G. A., Staffetti, E., Surinyac, J. (1999). Estructuras geométricas jerárquicas para la modelización de escenas 3D.
7. P5.js
8. Geeksforgeeks