

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN  
Algoritmos y Estructura de datos  
Práctica 4

Christian Néstor Barriga Marcapura  
Weimar Ccapatinta Huamani  
Roger Gutierrez Espinoza

Setiembre 12, 2022

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Ejercicios</b>	<b>2</b>
2.1	Funciones: build-kdtree, getHeight, generate-dot . . . . .	2
2.2	Implementación de closest point . . . . .	5
2.3	Función KNN . . . . .	6
<b>3</b>	<b>Implementación</b>	<b>7</b>

## 1 Introducción

En ciencias de la computación, es una estructura de datos de particionado del espacio que organiza los puntos en un Espacio euclídeo de  $k$  dimensiones. Los árboles kd son un caso especial de los árboles BSP.

Un árbol kd emplea sólo planos perpendiculares a uno de los ejes del sistema de coordenadas. Esto difiere de los árboles BSP, donde los planos pueden ser arbitrarios. Además, todos los nodos de un árbol kd, desde el nodo raíz hasta los nodos hoja, almacenan un punto. Mientras tanto, en los árboles BSP son las hojas los únicos nodos que contienen puntos (u otras primitivas geométricas). Como consecuencia, cada plano debe pasar a través de uno de los puntos del árbol kd.

## 2 Ejercicios

### 2.1 Funciones: build-kdtree, getHeight, generate-dot

Se realizó la implementación del algoritmo considerando las funciones mencionadas, tal como se muestra en la figura 1.

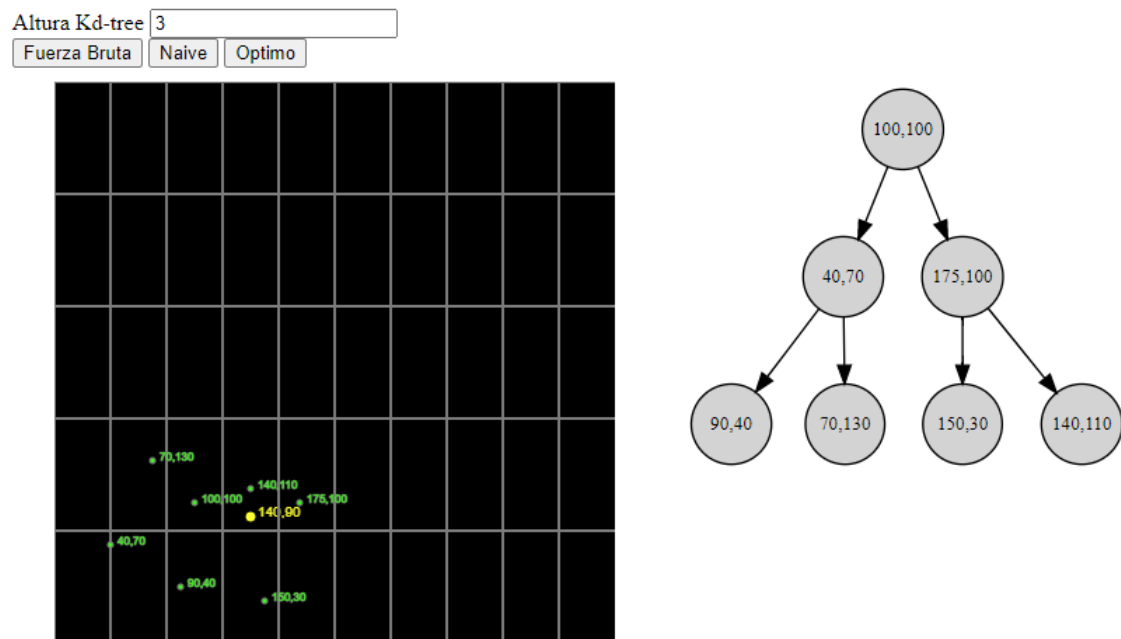


Figura 1 Funciones Kd Tree - Elaboración propia

Se implementaron tres funcionalidades, Fuerza bruta, Naive y optimo.

La fuerza bruta saca la distancia entre dos puntos y obtiene el menor.

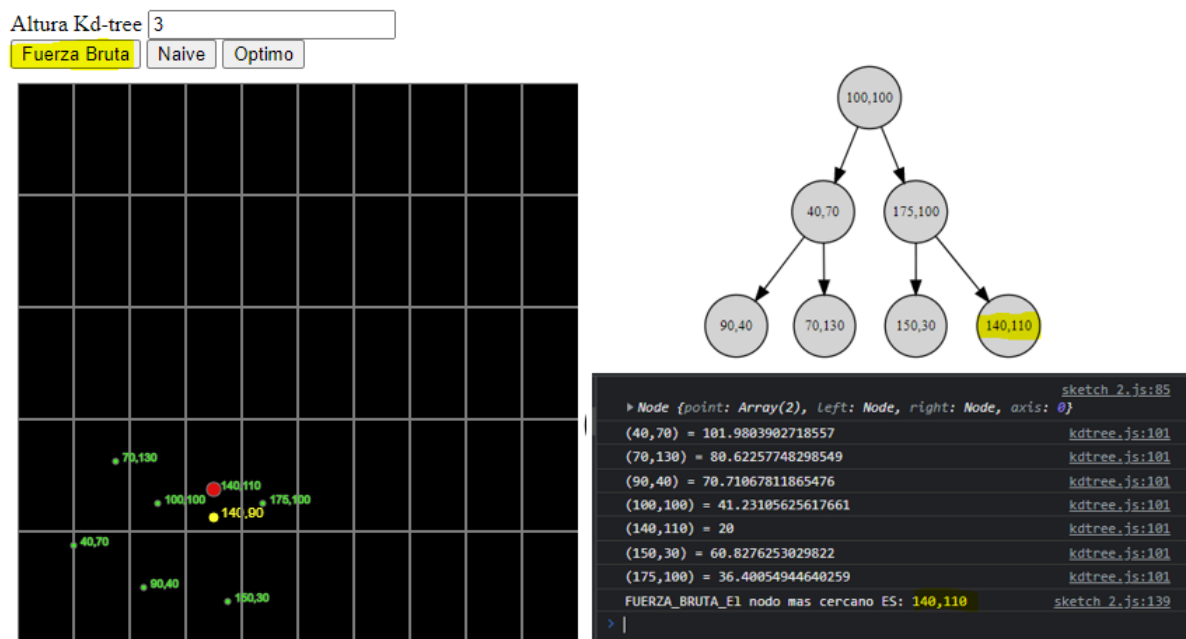


Figura 2 Fuerza bruta - Elaboración propia

Naive, sigue a los nodos comparando con los ejes, cuando estan cerca del límite, no revisa el otro nodo y tiende a equivocarse

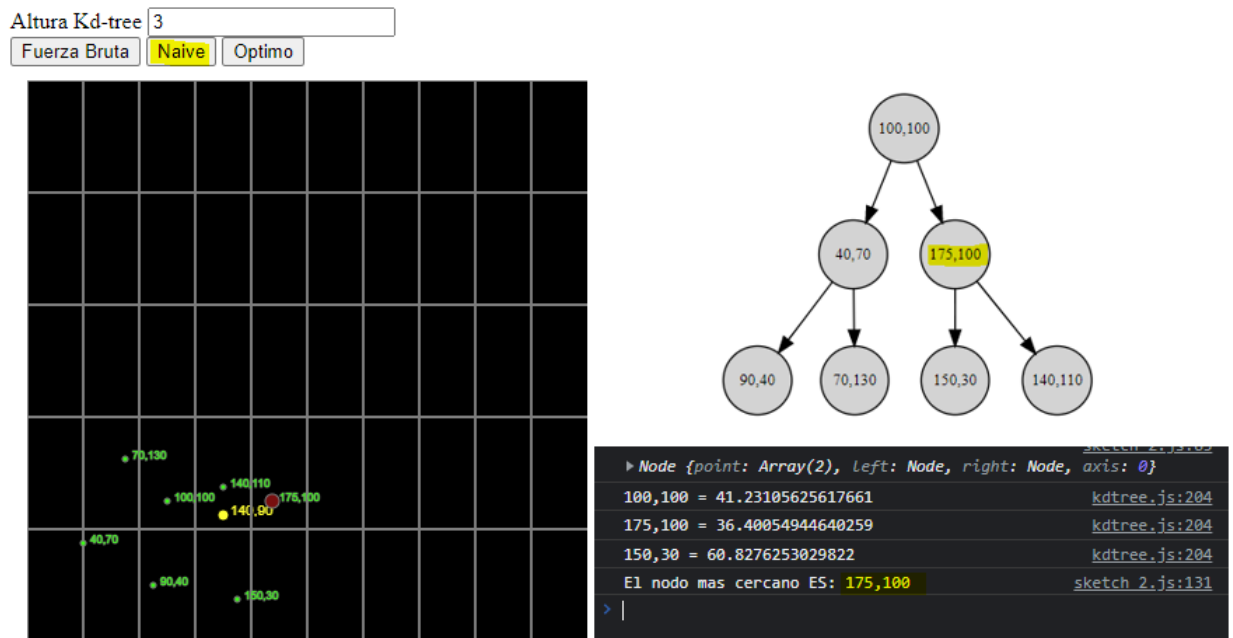


Figura 3 Función naive - Elaboración propia

Y finalmente Optimo

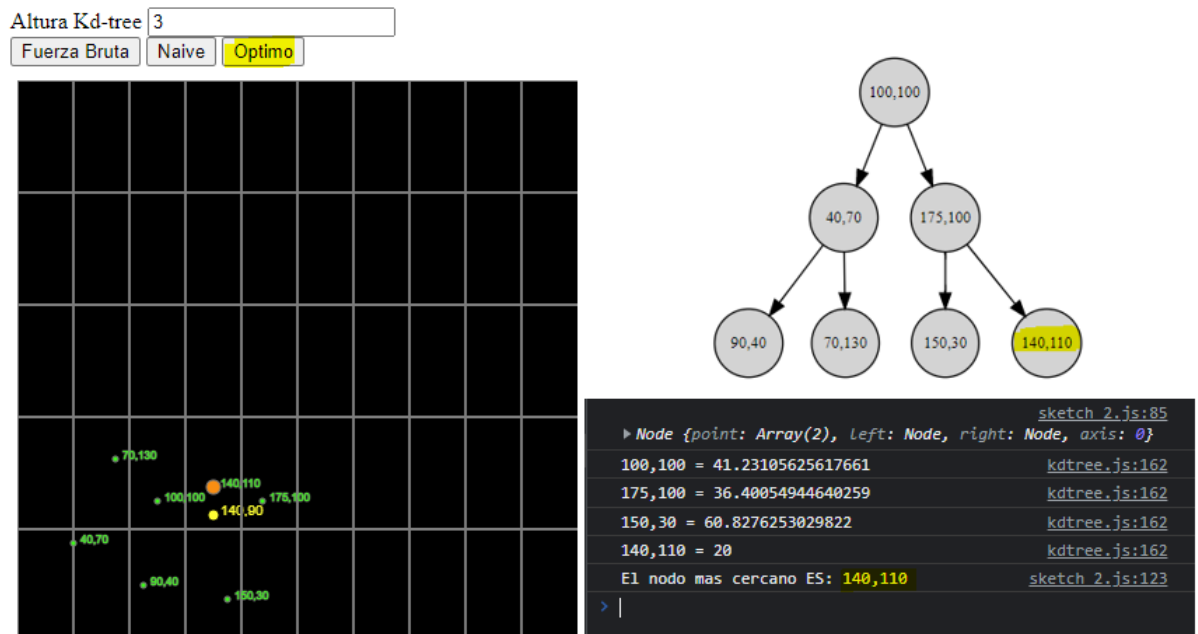


Figura 4 Búsqueda óptima - Elaboración propia

## 2.2 Implementación de closest point

En la siguiente figura se muestra como esta implementado el Closest point.

### Maestría en Ciencias de la Computación

Grupo 10 --> Kd Tree / KNN

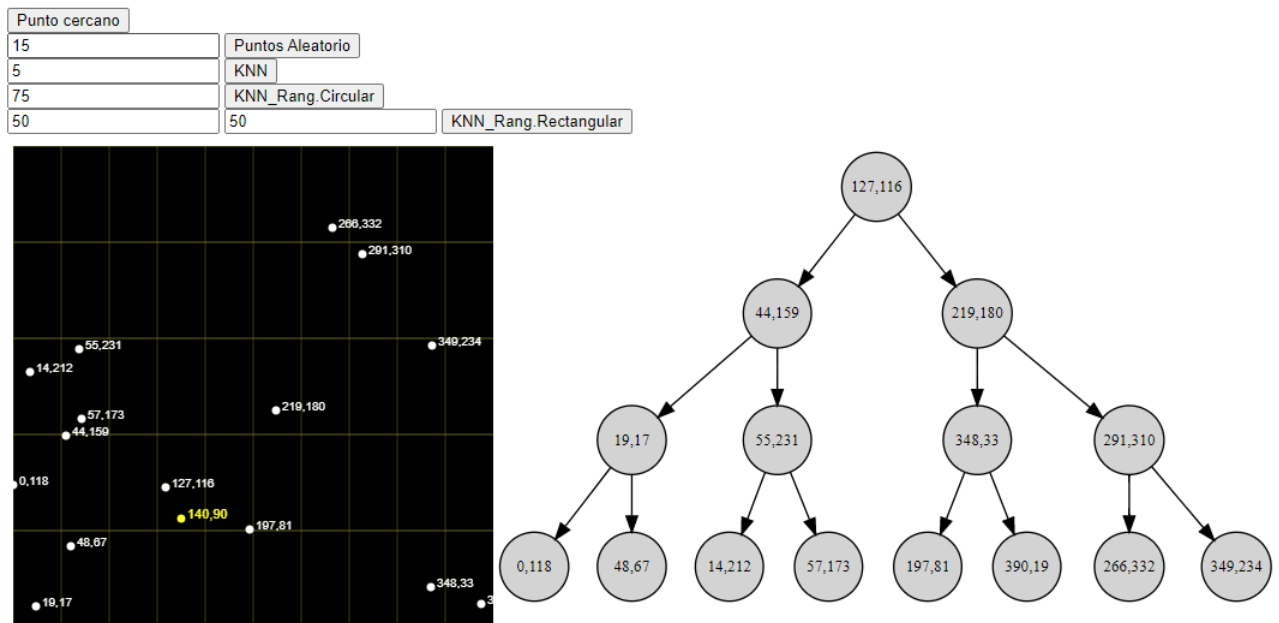


Figura 5 Closest point - Elaboración propia

## 2.3 Función KNN

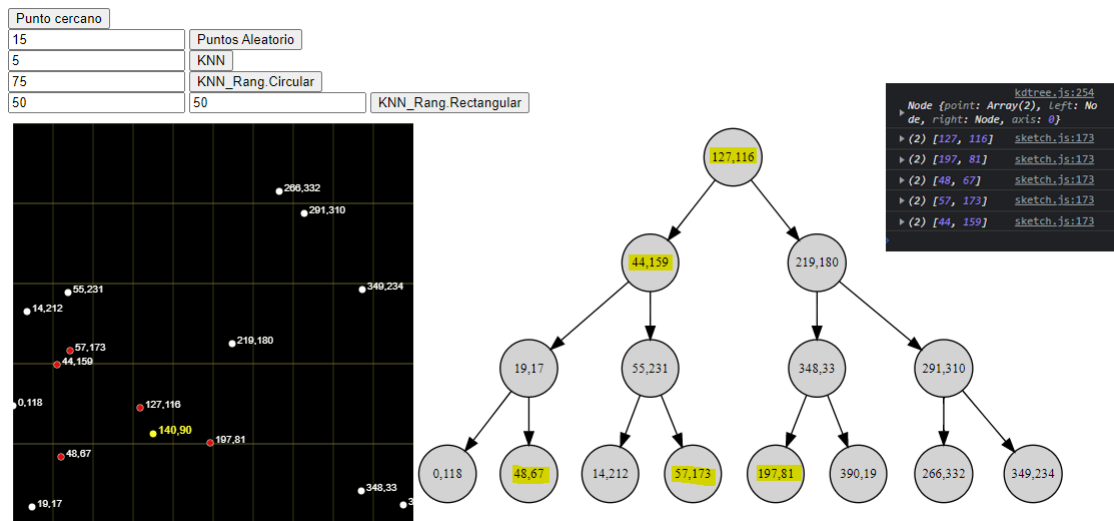


Figura 6 Función KNN - Elaboración propia.

Se consideraron dos tipos de range query:

Del tipo circular

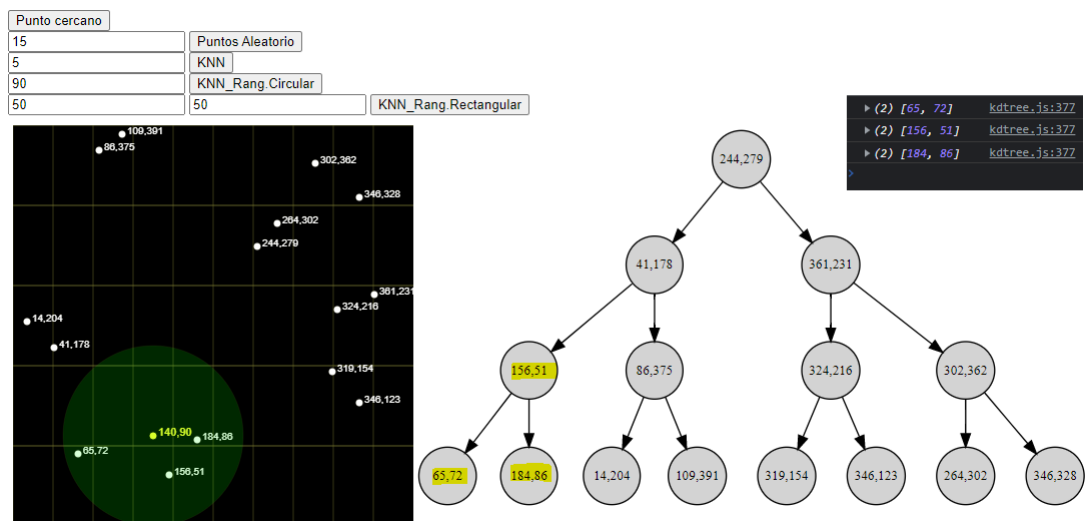


Figura 7 Función KNN query circular - Elaboración propia

Del tipo rectangular, realizando el sombreado del área que abarca cada figura.

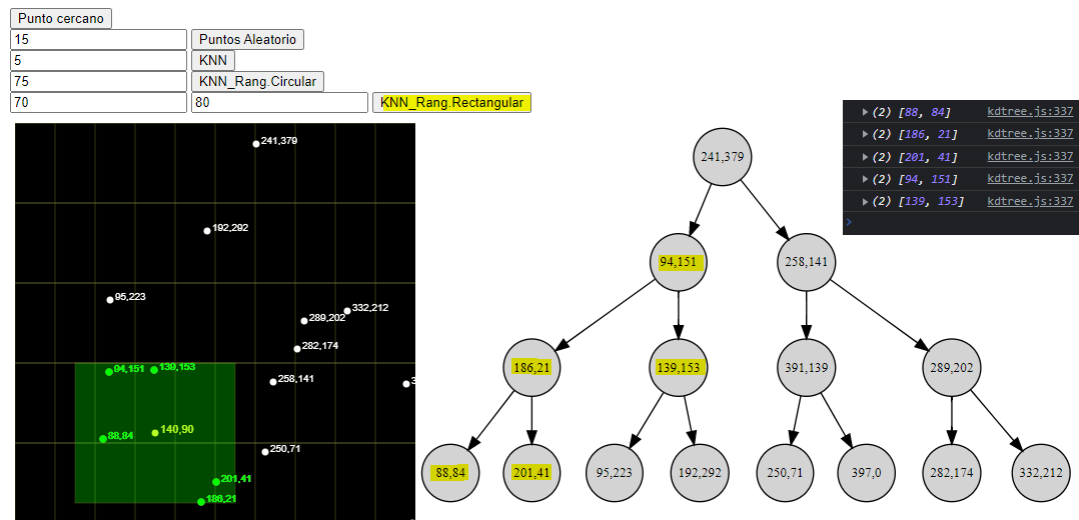


Figura 8 Función KNN query rectangular- Elaboración propia

### 3 Implementación

Se envía el siguiente enlace en Github, donde se encuentran los codigos elaborados y con las animaciones respectivas.

<https://github.com/weicap/Practica-4>