

## Práctica 06

DOCENTE	CARRERA	CURSO
MSc. Vicente Machaca Arceda	Escuela Profesional de Ciencia de la Computación	Estructura de Datos Avanzada

PRÁCTICA	TEMA	DURACIÓN
06	KD-Tree	2 horas

### 1. Competencias del curso

- Conocer e investigar los métodos de acceso multidimensional, métrico y aproximado.
- Analiza, diseña y propone soluciones utilizando estructuras de datos avanzadas.
- Comprende la importancia e impacto de los algoritmos estudiados y las nuevas propuestas.
- Aplica principios matemáticos para la solución de problemas.

### 2. Competencias de la práctica

- Comprende, implementa y aplica la estructura multidimensional *KD-Tree*.

### 3. Equipos y materiales

- Javascript, Python o C++
- Navegador Web
- Cuenta en Github
- IDE de desarrollo

### 4. Entregables

- Se debe elaborar un informe en Latex donde se responda a cada ejercicio de la Sección 5.
- En el informe se debe agregar un enlace al repositorio Github donde esta el código.
- En el informe se debe agregar el código fuente así como capturas de pantalla de la ejecución y resultados del mismo.
- El informe debe ser nombrado "GRUPO-X", donde "X" es el nombre del grupo (1A, 2A, 1B, etc.).
- El trabajo se presentará en dos hitos: en el primero se calificará la actividad 1 y 2, en el segundo hito se calificará lo restante.

## 5. Ejercicios

1. Implemente la función *closest\_point\_brute\_force* y *naive\_closest\_point* :

```
k = 2;

class Node{
    constructor(point, axis){
        this.point = point;
        this.left = null;
        this.right = null;
        this.axis = axis;
    }
}

function distanceSquared(point1, point2){
    var distance = 0;
    for (var i = 0; i < k; i++){
        distance += Math.pow((point1[i] - point2[i]), 2);
    }
    return Math.sqrt(distance);
}

function closest_point_brute_force(points, point){}
function naive_closest_point(node, point, depth = 0, best = null){}
```

2. Evalúe el resultado de las dos funciones implementadas anteriormente con este conjunto de datos:

```
var data = [
    [40,70],
    [70,130],
    [90,40],
    [110, 100],
    [140,110],
    [160, 100]
];
var point = [140,90]; //query
```

3. Evalúe el resultado de las dos funciones implementadas anteriormente con este conjunto de datos:

```
var data = [
    [40,70],
    [70,130],
    [90,40],
    [110, 100],
    [140,110],
    [160, 100],
    [150, 30]
];
var point = [140,90]; //query
```

4. Ahora implemente la función *closest\_point*, siguiendo las recomendaciones dadas por el docente:

```
function closest_point(node, point, depth = 0){}
```

5. Averigüe e implemente una función *KNN*, que retorne los  $k$  puntos más cercanos a un punto.

## 6. Rúbricas

Rúbrica	Cumple	Cumple con obs.	No cumple
<b>Informe:</b> El informe debe estar en Latex, con un formato limpio, buena presentación y redacción.	3	1.5	0
<b>Implementación:</b> Ha desarrollado todas las actividades solicitadas en la práctica.	12	6	0
<b>Presentación:</b> El alumno demuestra dominio del tema y conoce con exactitud cada parte de su trabajo.	5	2.5	0
<b>Errores ortográficos:</b> Por cada error ortográfico, se le descontará un punto.	-	-	-