

Taller en Sala 8 Dividir y Conquistar



Objetivos: 1. Usar un algoritmo de dividir y conquistar para resolver un problema adecuado. 2. Determinar un enfoque algorítmico apropiado para enfrentar un problema. 3. Identificar un ejemplo práctico para cada una de las estrategias de diseño de algoritmos.



Consideraciones: Lean y verifiquen las consideraciones de entrega,



Trabajo en Parejas



Mañana, plazo de entrega



Docente entrega plantilla de código en GitHub



Sí .cpp, .py o .java



No .zip, .txt, html o .doc



Alumnos entregan código sin comprimir GitHub



En la carpeta Github del curso, hay un código iniciado y un código de pruebas (tests) que pueden explorar para solucionar los ejercicios



Estructura del documento: a) Datos de *vida real*, b) *Introducción* a un problema, c) Problema a resolver, d) Ayudas. Identifiquen esos elementos así:

a)



b)



c)



d)



PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



En la vida real, en muchos videojuegos, se necesita ordenar los objetos en 3D según un criterio, usualmente la profundidad, para poder hacer una rasterización correcta; sobretodo, cuando hay capas que tienen transparencia. Para más información ver <https://bit.ly/2IR77MN>

Ejercicios a resolver

- 1 Una de las principales características del videojuego *Silent Hill 3* es la niebla, que para modelarla se utiliza la transparencia. Para poder renderizar texturas o efectos que usen transparencia, se necesita renderizar los elementos en orden del que está más atrás al que está más adelante para tener una visualización correcta.



El criterio de ordenamiento de los objetos 3D es una métrica que represente la profundidad. El problema de escoger el orden de los objetos a renderizar se reduce a un problema de ordenamiento de números.

- ▶ **Implementen el algoritmo de *Mergesort* y el algoritmo de *Quicksort* para ordenar un arreglo de enteros en orden ascendente.** ¿Qué diferencia hay entre un algoritmo y otro? ¿Cuál es mejor usar para el problema de ordenamiento de objetos transparentes según la profundidad?

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

Ayudas para resolver los Ejercicios

Ejercicio 1

Pág. 4



Ejercicio 1



Ejemplo 1: Si tenemos el arreglo [2, 10, 46, 1, 8] debe quedar [1, 2, 8, 10, 46].



Pista 1: Asuman que el arreglo dado tiene por lo menos un elemento.



Pista 2: Consideren que hay dos subproblema: El 1.1 es implementar *Mergesort* y el 1.2 es implementar *Quicksort*.

1.1

Mergesort

```
public static void mergesort(int[] a) {  
    // complete...  
}
```



Para la implementación les recomendamos dividir el algoritmo en dos: una parte para particionar recursivamente, y otra que una las particiones dejándolas ordenadas en el arreglo original.

Adicional a esto, utilicen índices que representen las particiones (en vez de crear arreglos más pequeños cada vez, haciendo un “particionado imaginario”), al igual que un arreglo auxiliar. Esto permite ahorrar memoria y hace más simple el proceso.

1.2

Quicksort

```
public static void quicksort(int[] a) {  
    // complete...  
}
```

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 2

Código ST0247



Esta vez también les recomendamos una implementación recursiva (el algoritmo, al igual que el anterior, naturalmente lo es). Este es un poco más simple; no se preocupen mucho con la elección del pivote (por ahora).



Ejercicio 1.1



Pista 1: Si tienen inconvenientes para entender el algoritmo les recomendamos el siguiente video: <https://bit.ly/19eOo5y>. No olviden activar los subtítulos en español en caso de que los necesite.



Ejercicio 1.2



Pista 1: Si tienen inconvenientes para entender el algoritmo les recomendamos el siguiente video: <https://bit.ly/1gDfAud>. No olviden activar los subtítulos en español en caso de que los necesite.

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 Ext. 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agenden una cita dando clic en la pestaña
-*Semana*- de <http://bit.ly/2gzVg10>