

EDA - Estructuras de Datos y Algoritmos

Experiencia

Los BST (Binary Search Trees) son unas estructuras de datos que te permiten implementar listas, tal que puedas realizar ciertas operaciones de forma más eficiente que con otras implementaciones, e.g., búsquedas o ordenamientos.

Competencias

Específicas:

- Conocimientos de estructuras de datos, como los Stacks, Queues, Linked Lists, y BST.

Transversales:

- Entender conceptos teóricos avanzados de Ciencia de la Computación

BKNs:

- Hacer un ejemplo 100% real de un curso más avanzado



Material

En **CANVAS** hay el Hackerrank con varias estructuras de datos y algoritmos a implementar, incluyendo "BST - search" que usaremos para el entregable



Problemas?: ¿Te está costando? ¿Por que no te pasas por **DISCORD** a buscar pareja de emociones? Las aventuras se viven mejor acompañado 😊



Entregable



Para entregar: Link en **CANVAS**

Para este entregable debes implementar la función "path" del problema "BST - search". Debes entregar una captura donde se vea que pasas el (único) testcase público.



Ejemplo 1

Testcase 0 ✓

Congratulations, you passed the sample test case.
Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

Your Output (stdout)

```
PATH 4
3
6
4
PATH 8
3
6
10
7
X
```

Expected Output

```
PATH 4
3
6
4
PATH 8
3
6
10
7
X
```



Rúbrica

Default view

Aa ID	# %	Descripción
R1	50%	BTS - search: Caso de Pasa el testcase donde en el BTS (PATH 4)
R2	50%	BTS - search: Caso de Pasa el testcase donde encuentra en el BTS (P
SUM		100%






Do It Yourself (DIY)



Para evaluar: Recuerda para que te evalúen los **DIY** y te den la nota bonus correspondiente, debes compartirllos en **DISCORD** siguiendo las instrucciones en **CANVAS**. ¡Así motivarás a los demás y podrás buscar sinergias!

 Default view

 Num	 Nivel	 DYI
<u>1</u>	Casual/Pro	Hacer el "BST - order" clase (sin mirar la solu
<u>2</u>	Casual/Pro	Aprender sobre alguna hash tables, red-black
<u>3</u>	Casual/Pro	Ver videos o leer sobre
<u>4</u>	Casual/Pro	Implementar alguna es como funciona
<u>5</u>	Casual/Pro	Aprender más sobre lo
<u>6</u>	Casual/Pro	Ojea alguno de los alg Algorithms: A Beginner
<u>7</u>	Casual/Pro	Investigar sobre alguna relacionada con estruc
...		... y cualquier otra idea



Exploración



IIC2133 - Estructuras de Datos y Algoritmos

Programa en la **CARPETA**

- ¿Cómo hacer que un código corra más rápido y de manera eficiente?
- ¿Cómo programar en lenguajes de bajo nivel? ¿Qué es bajo nivel?
- ¿Puedo calcular el desempeño de mi código y sus algoritmos?



IIC1253 - Matemática Discreta

Programa en la **CARPETA**

- ¿Cómo usar lógica para demostrar código?
- ¿Cómo modelar problemas computacionales con matemáticas?



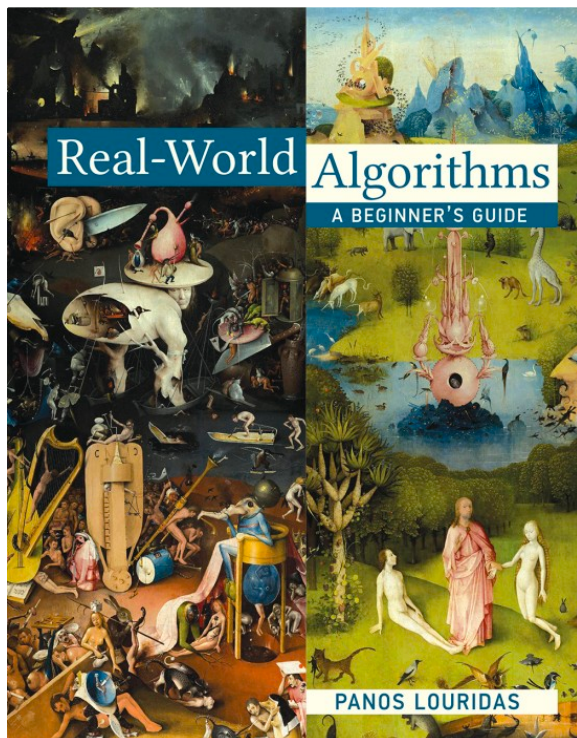
IIC2283 - Diseño y Análisis de Algoritmos

Programa en la **CARPETA**

- ¿Cómo diseñar un algoritmo cumpliendo ciertos requisitos?
- ¿Cuál es el potencial de un algoritmo y en qué áreas puede ser utilizado?



Bibliografía



Real-World Algorithms: A Beginner's Guide

Panos Louridas

The MIT Press (2017) [Link Bibliotecas UC](#)