

ÁREA 1. ALGORITMIA

SUBÁREA 1.2 ESTRUCTURAS DE DATOS

BIBLIOGRAFÍA DE LA GUÍA:

- Aho, Alfred, Hopcroft, John, E. y Ullman, Jeffrey. (1988). Estructuras de Datos y Algoritmos. Estados Unidos: Addison-Wesley Iberoamericana. 413 p.
- Cairo, Osvaldo y Guardati, Silvia. (2013). Estructuras de datos. 3a ed. México: Mc Graw Hill.
- Martínez, Román. y Quiroga, Elda. (2002). Estructuras de datos: referencia práctica con orientación a objetos. México: Thomson Learning.
- Thomas, Wu. (2008). Programación en Java: Introducción a la programación orientada a objetos. México: McGraw Hill.
- Wirth, Niklaus. (1987). Algoritmos y estructuras de datos. 2a ed. México: Prentice Hall. 306 p.

TEMAS IMPORTANTES:

Estructuras de Datos Simples:

Las estructuras de datos simples pueden ser desde variables (y los tipos de datos disponibles en algún lenguaje de programación), hasta el manejo de arreglos y sus diferentes tipos (simples, matrices, vectores, multidimensionales).

Para una referencia rápida de algunos de estos temas pueden consultar:

https://www.tutorialspoint.com/data_structures_algorithms/index.htm

Pilas y Colas:

Las pilas y colas son estructuras con una funcionalidad particular, y que pueden ser implementadas con arreglos o listas. Es importante saber cuando usarlas y cómo funcionan; así como los tipos que existen, particularmente de colas (normal, de prioridad). Se les puede preguntar que identifiquen cuál estructura se debe de usar en algún problema, o especificar cuál estructura se está usando en cierto código, etc.

Listas Enlazadas y sus Tipos:

Recordar las características, funcionamiento e implementación de las listas enlazadas y sus tipos (dobles, circulares, etc.). Se les puede preguntar cuál tipo de lista es más indicada para cierto problema, o que identifiquen la lista usada en cierto código, etc.

Árboles y sus Tipos:

Recordar las características, componentes, funcionamiento e implementación de los árboles y sus tipos (simples, balanceados, B, B+, binarios, etc.). También es importante recordar los algoritmos que son propios de los árboles, por ejemplo, la búsqueda binaria, etc. Se les puede preguntar cuál tipo de árbol es más indicado para cierto problema, que identifiquen el tipo de árbol que se está usando en cierto código, o el tipo de árbol que resultó después de ejecutar cierto código, etc. Hay un tipo especial, llamado **heap** o **montículo**, que se basa en una estructura tipo árbol. Esta estructura casi no la estudiamos, por lo que es bueno que la conozcan por si acaso:

https://www.tutorialspoint.com/data_structures_algorithms/heap_data_structure.htm

Grafos:

Recordar las características, componentes, funcionamiento e implementación de los grafos, sus tipos (dirigido, no dirigido, etc.), sus formas de representación (matriz de adyacencia, etc.). También es importante recordar los algoritmos que son propios de los grafos, por ejemplo, algoritmo de Floyd, Dijkstra, etc. Se les puede preguntar cuál tipo de grafo es más indicado para cierto problema, cuál grafo representa cierta información, que identifiquen el tipo de grafo que se está usando en cierto código, o el tipo de grafo que resultó después de ejecutar cierto código, etc.

Muchos de los algoritmos que se usan en árboles y grafos los pueden encontrar en el libro:

Neapolitan, R. Foundations of Algorithms. Jones & Bartlett Learning, 5th Edition, 2015.

Archivos:

Los archivos son estructuras de datos que se almacenan en medios diferentes a la memoria de la computadora. Hay que recordar las características de los archivos, sus tipos, su implementación, su funcionamiento, etc. Es importante que recuerden los diferentes algoritmos de organización de archivos, cómo se inserta, elimina, modifica información dentro del archivo. Algunos tipos de organizaciones son: Secuencial, indexada, por tablas Hash, árboles B/B+, entre otras. Una breve introducción la encuentran aquí:

<https://www.geeksforgeeks.org/file-organization-in-dbms-set-1/>

<https://www.geeksforgeeks.org/file-organization-in-dbms-set-4/?ref=lbp>

<https://www.geeksforgeeks.org/file-organization-in-dbms-set-3/?ref=lbp>

Ordenamiento y Búsqueda:

Una parte importante de las estructuras de datos son su ordenamiento y la búsqueda de información en estas. Hay que conocer y saber el funcionamiento de los diferentes algoritmos de búsqueda y ordenamiento de acuerdo con las estructuras de datos usadas. Por ejemplo, en

el caso de la búsqueda, los algoritmos más conocidos son el secuencial y binaria. Estos algoritmos se utilizan para buscar información en arreglos y listas principalmente. El funcionamiento de la búsqueda binaria se convierte de alguna manera en un árbol; sin embargo, si se considera que la información ya se encuentra en un árbol, se pueden tener otro tipo de algoritmos de búsqueda (por ejemplo, búsqueda primero en profundidad, o primero en amplitud).

Lo mismo puede suceder con los algoritmos de ordenamiento. Hay que recordar los algoritmos que se usan para ordenar datos en arreglos o listas, por ejemplo: Burbuja, Inserción, Selección, Quicksort, etc. Pero también puede haber métodos o técnicas un tanto diferentes para otras estructuras de datos, por ejemplo, se mencionó más arriba la organización de archivos tiene sus propios algoritmos. Las tablas Hash, por ejemplo, se usan tanto en arreglos como en archivos.

Se puede consultar el siguiente sitio para revisar de forma rápida las diferentes estructuras, métodos de ordenamiento y búsqueda (revisar el menú izquierdo):

https://www.tutorialspoint.com/data_structures_algorithms/index.htm

NOTAS FINALES: Una vez que se vieron todas las estructuras de datos de forma individual, se les podría preguntar acerca de cuál estructura utilizar para cierto problema, o identificar la estructura de acuerdo con la respuesta mostrada.

De forma similar a lo que se mencionó en la subárea 1.1, la guía no especifica un lenguaje particular, por lo que hay que familiarizarse con lenguajes clásicos como C/C++ y Java.