

SOLUCIONES ALGORÍTMICAS QUE REQUIERAN LA UTILIZACIÓN DE LOS MÉTODOS DE ORDENAMIENTO Y BÚSQUEDA

[Elemento competencia 2](#)[Guía de evidencia](#)[Bibliografía](#)

Desarrollo temático

Tabla de contenido

De clic en los siguientes enlaces para acceder a un contenido específico.

[Métodos de ordenación](#)[Métodos de búsqueda](#)

Métodos de Búsqueda

En el siguiente audio encontrarás la explicación del concepto de métodos de búsqueda. Te invitamos a escuchar atentamente y a tomar notas de los aspectos más importantes que allí se exponen: métodos búsqueda. Una vez finalizada la audición, puedes iniciar con el estudio de los cuatro métodos que se explican a continuación.

Búsqueda Secuencial

Se le denomina también lineal. Consiste en comenzar en el inicio de la lista y revisar a través de cada registro hasta encontrar el valor deseado o hasta el final de la lista.

Algoritmo:

- 🌟 Iniciamos con el primer registro
- 🌟 El valor buscado es igual al primer registro, de ser así se encontró el valor de lo contrario continúe en la búsqueda
- 🌟 Compara valor buscado con $[n+1]$ hasta llegar al fin de la lista o hasta encontrar el valor.

Ahondemos mas sobre este tipo de búsqueda, tomando información de Algoritmia.net, aquí encontraremos datos generales de esta búsqueda e información sobre el algoritmo:

<http://www.algoritmia.net/articles.php?id=32>

Búsqueda Binaria:

Se compara a la realización de la búsqueda de una palabra en un diccionario, el cual se abre cerca al inicio o al final dependiendo de la primera letra de la palabra que se busca. El buscador puede acertar encontrando la palabra en la página que abrió, sin embargo no es lo común, por lo cual se mueve hacia adelante o hacia atrás de la página que abrió dependiendo de la palabra que se busca. De igual manera se realiza con las listas ya ordenadas, se toma el centro de la lista, se comprueba que la clave coinciden con el valor del elemento central. Si no se encuentra la clave, se coloca uno en la mitad inferior o superior del elemento central de la lista, dependiendo del valor buscado.

Algoritmo

1. Se define una clave (número a buscar)
2. Calcula el punto central de la lista
3. Centro = total de números / 2 (división entera)
4. Compara el valor centro de la lista con la clave
5. Si la clave es igual al centro se ha encontrado el número, de lo contrario si la clave es mayor al centro se desechan todos los valores a la izquierda del centro o los valores bajos, por el contrario si la clave es menor al centro se desechan todos los valores a la derecha del centro o los valores altos.
6. Se obtiene una nueva sub-lista en la cual se realizan los pasos del 2 al 4.
7. El algoritmo termina porque se ha encontrado la clave o porque el valor bajo excede al alto y el algoritmo indica un fallo lo cual significa que no encontró el valor.

Veamos un ejemplo, nuevamente apoyados en las cartas de una baraja:

Tomado de: <http://youtu.be/oQw8vpCclW0>

Transformación de claves (Hash) Este tipo de búsqueda no requiere que se encuentre arreglado el array. Asigna a cada elemento un índice mediante la transformación del elemento, dicha transformación es la llamada Hash. La correspondencia más sencilla es la identidad, es decir al Número 0 se le asigna el índice 0, al num. 1 el índice 1 y así sucesivamente. La función hash depende del problema en el cual se desarrolle y de la finalidad, se pueden utilizar con números o cadenas.

Backtracking Se utilizan para encontrar soluciones a un problema, realizando una búsqueda sistemática, lo cual significa que se debe probar todo hasta encontrar una solución ó encontrar que no existe solución al problema, para esto se realiza una búsqueda en varias búsquedas parciales o sub-tareas.

Para fortalecer en estos dos tipos de búsqueda estás invitado a revisar Algoritmia.net, aquí encuentras información sobre cada una de ellas la manera como trabajan y algunos ejemplos:

<http://www.algoritmia.net/articles.php?id=32>

<http://www.algoritmia.net/articles.php?id=33>

Sobre estos dos temas se indagará con mayor profundidad en el curso Estructuras de datos II, al tener conocimientos previos de arboles y grafos.

Para fortalecer la información sobre métodos de ordenación y búsqueda se ofrece la lectura del capítulo 10: Algoritmos de ordenación y búsqueda. Programación en C: Metodología, algoritmos y estructura de datos. La cual encuentra en el siguiente vínculo.

<http://www.mcgraw-hill.es/bcv/guide/capitulo/8448198441.pdf>

De igual manera, el siguiente video en ingles realizado por los profesore Eric Grimson y John Guttag, para el curso de introducción a la ciencia de la computación y la programación del instituto de Tecnología de Massachusetts

Tomado de: <http://youtu.be/OK34iKj6I9Y>

Bien, hasta el momento hemos estudiado estructuras lineales estáticas por medio de vectores y matrices, definiéndolas desde el inicio en su tamaño y tipo. Hemos estudiado también sobre métodos de ordenación y búsqueda dentro de ellos y se nos ha permitido encontrar las grandes bondades que tienen este tipo de estructuras.

Sin embargo, al ser estructuras estáticas NO permite variar en su tamaño, entonces nos preguntamos ¿qué sucede si necesitamos realizar un cambio en la estructura de ingreso, por ejemplo, qué pasa si debemos realizar una estructura en la cual día a día se ingresan nuevos registros como las partidas de nacimiento de los niños de una región? En este caso no podríamos utilizar listas estáticas porque no conocemos el tamaño del listado, en otras palabras: no sabemos cuántos niños nacerán de aquí en adelante, ni tampoco podríamos crear una lista muy grande porque generaría demasiado espacio en memoria que sería subutilizado. Es por esto que en estructuras de datos encontramos las listas enlazadas o estructuras dinámicas son ellas quienes permiten flexibilidad al momento de realizar modificaciones como la ejemplificada. Esos son pues, algunos de los temas que trabajaremos en la última unidad de este curso.

A continuación encontrarás las actividades que te permiten fortalecer los conocimientos adquiridos hasta el momento y llevarlos a la práctica.

