ALGORITMOS DE ORDENAMIENTO

**Ordenamiento por inserción:** El algoritmo propone tomar un elemento del arreglo y colocarlo en la posición correcta con respecto a los demás elementos ya ordenados. Inicialmente se tiene solo un elemento ordenado, luego se toma el elemento K+1 y se compara con todos los elementos ya ordenados, deteniéndose cuando se encuentra con un elemento menor, en esa posición es insertado y los demás elementos son desplazados una posición a la derecha.

**Método de la burbuja:** en este método se comparan los elementos de a parejas, el elemento n con el elemento n+1, si cumplen con la condición de que n < n+1 se Avanza y se comparan los siguientes dos elementos, si la condición no se cumple intercambian sus posiciones. Así el elemento más pequeño flota hasta el comienzo como si fuera una burbuja.

**Ordenamiento por selección:** en este método se hace una primera recorrida para buscar el menor y colocarlo al principio, luego se hace otra recorrida empezando desde la segunda posición y se busca nuevamente el menor. Así el procedimiento continúa hasta que todos los elementos estén ordenados.

**Quick-sort:** es el método más rápido de ordenamiento, originalmente es un método recursivo, pero hay versiones iterativas para mejorar su rendimiento.

El método consiste en elegir un elemento del arreglo al que llamaremos pivote y reubicar los demás elementos de la lista de tal forma que a la izquierda del pivote queden los más chicos y a la derecha los más grandes. De esta manera la lista quedara dividido en 2 sub listas. A continuación, se repite el mismo proceso en las sub listas de manera recursiva hasta que los elementos estén ordenados.

ALGORITMOS DE BÚSQUEDA

**Búsqueda secuencial:** Este algoritmo recorre secuencialmente una lista hasta encontrar el elemento buscado o hasta terminar de recorrer la lista sin encontrar el elemento buscado. El proceso se realiza con un bucle “for”, analizando cada elemento de la lista, uno por vez.

**Búsqueda binaria:** Se utiliza cuando tenemos listas ya ordenadas y es ventajoso para cuando estamos trabajando con listas con gran cantidad de elementos ya que el proceso es más corto.

El procedimiento consiste en definir el punto medio del segmento a analizar para la búsqueda. Luego, se analiza si el elemento buscado se encuentra en el medio o no, y, de no ser así, se analiza si el elemento buscado es mayor o menor al valor central. Si es menor, se analiza el segmento de la izquierda, y si no lo es, se analiza el segmento de la derecha. Esto se vuelve a repetir sucesivamente con cada subsegmento hasta encontrar el valor buscado o hasta terminar de recorrer la lista.

ALGORITMOS DE RECORRIDO

Para visualizar o consultar los datos almacenados en un árbol se necesita recorrer el árbol o visitar los nodos del mismo. Al contrario de las listas, los árboles binarios no tienen realmente un primer valor, un segundo valor, tercer valor, etc. El recorrido de árboles binarios requiere que cada nodo sea visitado solo una vez para ello existen dos enfoques distintos:

**Recorrido en profundidad:** se realiza un camino desde el nodo raíz, hacia el primer hijo y hasta el descendiente más lejano de ese hijo antes de proseguir al segundo hijo. Para saber como regresamos, vamos guardando los nodos ya visitados en una estructura de pila, por ello normalmente este método se programa de modo recursivo.

**Recorrido en anchura:** el proceso se realiza horizontalmente, desde la raíz a todos sus hijos, a continuación, los hijos de sus hijos y así sucesivamente. En otras palabras, se procesa todo un nivel antes de pasar al siguiente. Para poder saber que vértices visitar se utiliza una cola.