

ALGORITMOS FUNDAMENTALES Y ARRAY

1- Explique brevemente en qué consisten los siguientes Algoritmos Fundamentales.

Algoritmos de Ordenación:

- Ordenamiento por Inserción:

Inicialmente se tiene un solo elemento, que obviamente es un conjunto ordenado. Después, cuando hay k elementos ordenados de menor a mayor, se toma el elemento $k+1$ y se compara con todos los elementos ya ordenados, deteniéndose cuando se encuentra un elemento menor (todos los elementos mayores han sido desplazados una posición a la derecha) o cuando ya no se encuentran elementos (todos los elementos fueron desplazados y este es el más pequeño). En este punto se inserta el elemento $k+1$ debiendo desplazarse los demás elementos.

- Algoritmo de la burbuja

Consiste en comparar cada elemento de la lista con el siguiente (por parejas), si no están en el orden correcto, se intercambian entre sí sus valores. El valor más pequeño flota hasta el principio de la lista

- Ordenamiento por selección

Se se recorre toda la lista buscando el menor de todos los elementos, una vez terminada la recorrida el menor elemento se coloca al inicio de la lista recorrida. En la siguiente iteración se recorre nuevamente la lista pero comenzando en el segundo elemento (ya que al haber insertado el menor encontrado al inicio ya lo consideramos ordenado). El procedimiento continúa hasta que el último elemento recorrido es el menor de su subconjunto.

- Algoritmo Quick-Sort

Se elige un elemento de la lista de elementos a ordenar, al que llamaremos pivote., luego colocamos a un lado todos los menores que el, y al otro lado los mayores que él, si los elementos son iguales al pivote es indiferente si se colocan a izquierda o derecha

La lista queda separada en dos sublistas, una formada por los elementos a la izquierda del pivote, y otra por los elementos a su derecha. Luego deberemos repetir este proceso de forma recursiva para cada sublista mientras éstas contengan más de un elemento. Una vez terminado este proceso todos los elementos estarán ordenados.

Algoritmos de Búsqueda:

- Búsqueda Secuencial

Se recorre secuencialmente la lista desde un elemento al siguiente, comenzando en la primera posición de la lista y se detiene cuando encuentra el elemento buscado o bien se alcanza el final de la lista sin haberlo encontrado.

- Búsqueda Binaria

Este algoritmo se utiliza cuando disponemos de una lista ordenada, ya que podemos ir disminuyendo el espacio de búsqueda a segmentos menores a la lista original. La idea es no buscar en aquellos segmentos de la lista donde sabemos que el valor seguro que no puede estar, considerando que la lista esta ordenada.

Podemos comenzar la búsqueda considerando la lista completa y evaluar un valor central de la misma, es probable que ese valor central no sea el buscado, pero podemos ver si ese valor central es mayor o menor al elemento buscado, para luego elegir en cual de los dos segmentos buscamos luego, logrando reducir la cantidad de elementos a buscar a la mitad. Si proseguimos con este procedimiento y continuamos buscando el valor central de cada segmento obtenido, podemos ir reduciendo cada vez más el espacio de búsqueda hasta llegar al elemento buscado, si es que existe en la lista.

Algoritmos de Recorrido:

- Recorrido de Profundidad

El proceso exige un camino desde el nodo raíz a través de un hijo, al descendiente más lejano del primer hijo antes de proseguir a un segundo hijo. En otras palabras, en el recorrido en profundidad, todos los descendientes de un hijo se procesan antes del siguiente hijo. Para saber cómo regresarnos, vamos guardando los nodos visitados en una estructura de pila. Es por esto que se acostumbra programar esta búsqueda de forma recursiva, con lo que el manejo de la pila lo realiza el lenguaje de programación utilizado.

- Recorrido de Anchura

El proceso se realiza horizontalmente desde el raíz a todos sus hijos, a continuación, a los hijos de sus hijos y así sucesivamente hasta que todos los nodos han sido procesados. En otras palabras, en el recorrido en anchura, cada nivel se procesa totalmente antes de que comience el siguiente nivel. Para poder saber qué vértices visitar, utilizamos una cola.