



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Materia: Estructura de Datos

Fase 2

Actividad 1

Maestro: NOEL ALEJANDRO HORTIALES CORONA

Alumnos: Dante Rodríguez Rodríguez 1899372

Castillo López Alfredo Ventura 2000505 IAS

Sánchez Alvarado Fabritzio Javier 2059273 IAS

Loredo Galindo Esteban Alessandro 2052205 ITS

ORDENAMIENTO DE BURBUJA

El método de ordenamiento de burbuja es un algoritmo de ordenamiento común utilizado en programación para ordenar una lista de elementos en orden ascendente o descendente. El algoritmo funciona comparando pares de elementos adyacentes en la lista y realizando intercambios si están en el orden incorrecto. Este proceso se repite hasta que no se realizan más intercambios, lo que indica que la lista está ordenada.

Las principales características del método de ordenamiento de burbuja son su simplicidad y facilidad de implementación. Sin embargo, es relativamente ineficiente para grandes conjuntos de datos y tiene una complejidad de tiempo de $O(n^2)$, lo que significa que el tiempo de ejecución aumenta exponencialmente con el tamaño de la lista de elementos.

Para utilizar el método de ordenamiento de burbuja, se requiere una lista de elementos a ordenar y un algoritmo que implemente la comparación y el intercambio de elementos adyacentes. Además, es importante considerar el tamaño de la lista de elementos y evaluar si el método de ordenamiento de burbuja es el más eficiente para el conjunto de datos en cuestión.

La notación Big O para el método de ordenamiento de burbuja es $O(n^2)$, lo que significa que el tiempo de ejecución aumenta cuadráticamente con el tamaño de la lista de elementos. La gráfica correspondiente a esta notación muestra una curva de crecimiento exponencial a medida que aumenta el tamaño de la lista de elementos.

ORDENAMIENTO CLAM SHELL

El método de ordenamiento Clam Shell es un algoritmo de ordenamiento que utiliza la técnica de inserción, pero con una estrategia de división en subconjuntos. Este método se basa en la comparación y el intercambio de elementos separados por una cierta brecha, que se reduce en cada iteración hasta que la lista de elementos está ordenada.

El algoritmo comienza dividiendo la lista en subconjuntos de elementos separados por una brecha inicialmente grande. Luego, se ordenan cada uno de los subconjuntos utilizando la técnica de ordenamiento por inserción. En cada iteración, la brecha se reduce y los subconjuntos se ordenan nuevamente hasta que la brecha alcanza un valor mínimo y todos los elementos están ordenados.

Las principales características del método de ordenamiento Clam Shell son su eficiencia para listas de elementos parcialmente ordenados y su adaptabilidad a diferentes tamaños de lista. Sin embargo, su complejidad de tiempo no es tan buena como otros métodos de ordenamiento, y puede ser de $O(n^2)$ en el peor de los casos.

Para utilizar el método de ordenamiento Clam Shell, se requiere una lista de elementos a ordenar, una brecha inicial y un algoritmo que implemente la comparación y el intercambio de elementos separados por la brecha. Es importante evaluar el tamaño y el grado de ordenamiento de la lista de elementos para determinar la brecha inicial adecuada y obtener una eficiencia óptima del algoritmo.

La notación Big O para el método de ordenamiento Clam Shell varía dependiendo de la brecha inicial y de cómo se reduce en cada iteración. En el peor de los casos, puede ser de $O(n^2)$, mientras que en el mejor de los casos puede ser de $O(n \log n)$. La gráfica correspondiente a esta notación muestra una curva de crecimiento cuadrático en el peor de los casos y una curva de crecimiento logarítmico en el mejor de los casos.

ORDENAMIENTO QUICK SORT

El método de ordenamiento Quick Sort es un algoritmo de ordenamiento basado en la técnica de "divide y vencerás". Este método se basa en la selección de un elemento pivote en la lista de elementos a ordenar y en la partición de la lista en dos subconjuntos: uno con elementos menores o iguales al pivote, y otro con elementos mayores al pivote. Luego, se aplica recursivamente el mismo proceso a cada subconjunto hasta que la lista completa esté ordenada.

El algoritmo de Quick Sort selecciona un elemento pivote en la lista y lo utiliza para dividir la lista en dos subconjuntos, uno con elementos menores o iguales al pivote, y otro con elementos mayores al pivote. Luego, se aplica recursivamente el mismo proceso a cada subconjunto hasta que la lista completa esté ordenada.

Las principales características del método de ordenamiento Quick Sort son su eficiencia para listas de elementos grandes y su facilidad de implementación. Sin embargo, su complejidad de tiempo depende del pivote seleccionado y puede ser de $O(n^2)$ en el peor de los casos.

Para utilizar el método de ordenamiento Quick Sort, se requiere una lista de elementos a ordenar, un algoritmo para seleccionar el elemento pivote y un algoritmo para particionar la lista en dos subconjuntos. Es importante evaluar el tamaño y el grado de desordenamiento de la lista de elementos para seleccionar el pivote adecuado y obtener una eficiencia óptima del algoritmo.

La notación Big O para el método de ordenamiento Quick Sort es de $O(n \log n)$ en el mejor de los casos y de $O(n^2)$ en el peor de los casos. La gráfica correspondiente a esta notación muestra una curva de crecimiento logarítmico en el mejor de los casos y una curva de crecimiento cuadrático en el peor de los casos.

ORDENAMIENTO DE BUSQUEDA SECUENCIAL

El método de búsqueda secuencial, también conocido como búsqueda lineal, es un algoritmo de búsqueda que recorre secuencialmente cada elemento de una lista para encontrar un valor específico. Este método es útil para listas pequeñas y desordenadas.

El algoritmo de búsqueda secuencial recorre cada elemento de la lista en orden y compara el valor del elemento actual con el valor buscado. Si encuentra el valor buscado, devuelve la posición del elemento en la lista. Si termina de recorrer la lista sin encontrar el valor buscado, devuelve un valor indicando que el valor no está presente en la lista.

Las principales características del método de búsqueda secuencial son su sencillez y facilidad de implementación, pero su complejidad de tiempo puede ser alta en listas grandes. La complejidad de tiempo del algoritmo de búsqueda secuencial es de $O(n)$, donde n es el tamaño de la lista.

Para utilizar el método de búsqueda secuencial, se requiere una lista de elementos y un valor buscado. El algoritmo recorre la lista desde el primer elemento hasta el último, comparando cada elemento con el valor buscado.

La notación Big O para el método de búsqueda secuencial es de $O(n)$, donde n es el tamaño de la lista. La gráfica correspondiente a esta notación muestra una curva de crecimiento lineal en función del tamaño de la lista.

ORDENAMIENTO DE BUSQUEDA BINARIA

El método de búsqueda binaria, también conocido como búsqueda dicotómica, es un algoritmo de búsqueda utilizado para encontrar un valor específico en una lista ordenada. Este método es útil para listas grandes y ordenadas.

El algoritmo de búsqueda binaria compara el valor buscado con el valor central de la lista. Si el valor buscado es igual al valor central, se devuelve la posición del valor central en la lista. Si el valor buscado es mayor que el valor central, la búsqueda continúa en la mitad derecha de la lista. Si el valor buscado es menor que el valor central, la búsqueda continúa en la mitad izquierda de la lista. Este proceso se repite hasta que se encuentra el valor buscado o se determina que no está presente en la lista.

Las principales características del método de búsqueda binaria son su eficiencia en listas grandes y ordenadas, pero no es útil para listas pequeñas o desordenadas. La complejidad de tiempo del algoritmo de búsqueda binaria es de $O(\log n)$, donde n es el tamaño de la lista.

Para utilizar el método de búsqueda binaria, se requiere una lista ordenada de elementos y un valor buscado. El algoritmo divide repetidamente la lista en mitades y descarta la mitad donde el valor buscado no puede estar, hasta que se encuentra el valor buscado o se determina que no está presente en la lista.

La notación Big O para el método de búsqueda binaria es de $O(\log n)$, donde n es el tamaño de la lista. La gráfica correspondiente a esta notación muestra una curva de crecimiento logarítmico en función del tamaño de la lista.

Referencias

- GeeksforGeeks. (n.d.). Bubble Sort. Recuperado el 29 de abril de 2023, de <https://www.geeksforgeeks.org/bubble-sort/>
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3ra ed.). MIT Press.
 - "Shell Sort" en GeeksforGeeks: <https://www.geeksforgeeks.org/shell-sort/>
 - Bolaños, R. (2011). Algoritmos y estructuras de datos. México: Pearson Educación.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). Cambridge, MA: MIT Press.
- GeeksforGeeks. (2021). Linear Search. Recuperado el 29 de abril de 2023, de <https://www.geeksforgeeks.org/linear-search/>
- TutorialsPoint. (s.f.). Data Structures - Linear Search. Recuperado el 29 de abril de 2023, de https://www.tutorialspoint.com/data_structures_algorithms/linear_search_algorithm.htm
- GeeksforGeeks: <https://www.geeksforgeeks.org/binary-search/>