

Investigación

- Algoritmos de ordenamiento

Los algoritmos de ordenamiento son un conjunto de instrucciones que toman una matriz o lista como entrada y organizan los elementos en un orden particular.

Debido a que la clasificación a menudo reduce la complejidad de un problema, es un algoritmo importante en Ciencias de la Computación. Los algoritmos de ordenamiento tienen aplicaciones directas en algoritmos de búsqueda, algoritmos de bases de datos, métodos de dividir y conquistar, algoritmos de estructura de datos y muchos más.

Algunos de los algoritmos de ordenamiento más comunes son:

- Selection Sort
- Bubble Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Heap Sort
- Counting Sort
- Radix Sort
- Bucket Sort

La complejidad algorítmica representa la cantidad de recursos (temporales) que necesita un algoritmo para resolver un problema y por tanto permite determinar la eficiencia de dicho algoritmo.

Cualquier algoritmo de ordenación basado en comparaciones debe hacer al menos $n \log_2 n$ comparaciones para ordenar la matriz de entrada. Para esta situación el algoritmo que implementamos fue merge sort que tiene una complejidad de $O(n \log n)$.

- Algoritmos de búsqueda

Un algoritmo de búsqueda resuelve como su nombre lo dice un problema de búsqueda, es decir, recupera la información almacenada dentro de alguna estructura de datos, o la calculada en el espacio de búsqueda de un dominio de problema.

Existen diferentes algoritmos de búsqueda. Dos de ellos son búsqueda secuencial y búsqueda binaria.

Para esta situación se implementó una búsqueda binaria, que tiene una complejidad máxima de $O(\log n)$. Esto quiere decir que el número máximo de operaciones necesarias para encontrar el objetivo de búsqueda es una función logarítmica del tamaño del espacio de búsqueda.

Reflexión

Durante las primeras 5 semanas de la clase TC1031.10, aprendimos distintos algoritmos de ordenamiento y búsqueda.

Después de esta actividad comprendí la importancia que tienen estos y que no todos tienen la misma eficiencia, por lo que dependiendo la situación es el algoritmo que debemos emplear.

La situación presentada en esta actividad es algo muy común de ver en la vida real, de aquí radica la importancia que tienen los algoritmos de ordenamiento y búsqueda.

La situación presentada nos habla de un registro que cuenta con un formato, pero no está ordenado, para esta situación nosotros primero ordenamos el registro y después de acuerdo al rango de fecha y hora que ingresado por el usuario fue que se realizó una búsqueda en el registro previamente ordenado.

Es importante el conocer y saber cómo funcionan los algoritmos de ordenamiento y búsqueda, pero es aún más importante el emplearlos en situaciones de la vida real a las que con frecuencia nos vamos a enfrentar como la que fue presentada.

Ahora entiendo que si no empleáramos los algoritmos de búsqueda y ordenamiento tendríamos que verificar cada uno de los elementos en el registro de forma individual para ver si coincide con lo que el usuario pide. No hay problema si el conjunto de datos es pequeño, pero como pasa regularmente se tienen grandes conjuntos de datos, y de no emplear los algoritmos este proceso llevaría mucho tiempo.

Referencias

FreeCodeCamp (18 de junio, 2020). Sorting Algorithms Explained. Recuperado de <https://www.freecodecamp.org/news/sorting-algorithms-explained/>

Mehlhorn, K. (2013). Data structures and algorithms 1: Sorting and searching (Vol. 1). Springer Science & Business Media.

FreeCodeCamp (13 de diciembre, 2019). Search Algorithms Explained with Examples in Java, Python, and C++. Recuperado de <https://www.freecodecamp.org/news/search-algorithms-explained-with-examples-in-java-python-and-c/>

Marisol, H. H., Rodolfo, M. S., y Raúl, S. G. F. Análisis de algoritmos de ordenamiento y búsqueda. Recuperado de http://ri.uaemex.mx/bitstream/handle/20.500.11799/69985/secme-3691_1.pdf?sequence=1