COLOMBIA POTENCIA DE LA VIDA

TIC

Complejidad **algorítmica**

Lección 3

TALENTO TECH

AZ | PROYECTOS EDUCATIVOS

Universidad Tecnológica de Pereira

Desarrollo teórico de la sesión: 5 horas

PLANTEAMIENTO DE LA SESIÓN
En esta sesión se detallarán las tareas del análisis de datos, explorando las tareas fundamentales que los científicos de datos realizan para extraer información valiosa de conjuntos de datos.
Una vez finalizada la actividad previa, el siguiente paso es exponer el concepto de complejidad algorítmica y la notación Big O. Una vez finalizada la explicación teórica se va a finalizar con un taller acerca de los conceptos tratados en la lección.

Algoritmo

Antes de entrar en los conceptos de complejidad algorítmica y notación Big O se tiene que definir ¿qué es un algoritmo?, y ¿cuáles son sus cualidades principales?

Un algoritmo es una sucesión de pasos bien definidos, no ambiguos y finitos para resolver un problema. Este debe ser preciso y determinístico, es decir, debe dar el mismo resultado ante una entrada igual. Debe ser eficiente, esto se consigue cuando consume menos recursos en unidades de tiempo, espacio en memoria y capacidad de cómputo consumida.

Se ha definido lo que es un algoritmo, ahora se puede hablar del concepto:

Complejidad algorítmica

Complejidad algorítmica

Es un concepto que se centra en medir y analizar la eficiencia de los algoritmos en función del tamaño de una entrada. Se mide en unidades de tiempo y los recursos necesarios para la ejecución, la complejidad crece en la medida en que se necesite más tiempo y recursos del sistema para ejecutar un algoritmo. Para medir esta complejidad se usa la notación Big O, la cual la representa de manera general y asintótica.

En términos generales, la complejidad algorítmica hace referencia a la eficiencia de un algoritmo y cómo esta se ve afectada por un aumento en el tamaño de los datos de entrada. Un algoritmo eficiente tiene una complejidad menor, lo que quiere decir que su rendimiento se ve menos afectado por conjunto de datos más grandes.

Notación Big O.

La notación Big-O es una herramienta que nos ayuda a entender cómo se comportan los algoritmos en relación con el tamaño de los datos que manejan. En términos generales, nos proporciona una indicación de la eficiencia y escalabilidad de un algoritmo.

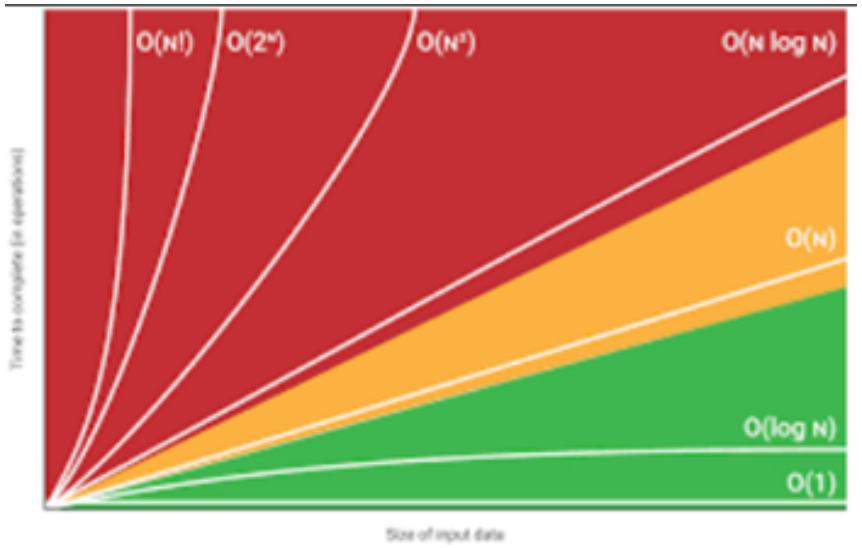
Por ejemplo, una función que busca un elemento en una lista. Si la complejidad es O(1), significa que la operación siempre toma aproximadamente la misma cantidad de tiempo, sin importar cuántos elementos tenga la lista. Por otro lado, si la complejidad es O(n), el tiempo de ejecución crece linealmente con el tamaño de la lista. Algunas notaciones Big O comunes incluyen:

Ejemplo

Algunas notaciones Big O comunes incluyen:

- **O(1):** constante, la operación no depende del tamaño de los datos.
- **O(n):** lineal, el tiempo de ejecución es proporcional al tamaño de los datos.
- **O(log n):** logarítmica, asociada con algoritmos que "trocean" el problema.
- **O(n log n):** similar a O(n log n), pero con procesamiento adicional.
- **O(n^2):** cuadrática, típica en algoritmos que requieren iteración por todos los elementos en cada uno de los elementos a procesar.
- **O(2^n):** exponencial, donde la complejidad se duplica con cada elemento añadido.
- **O(n!):** Combinatoria, generalmente asociada con algoritmos fallidos que utilizan fuerza bruta.

La elección entre algoritmos similares se basa en seleccionar aquellos con una complejidad asintótica menor para garantizar un mejor rendimiento a medida que los datos crecen.



En la figura se puede observar gráficamente en cada una de las notaciones más comunes, iniciando por los que están en la franja verde, las cuales son altamente eficientes sin importar el tamaño de la entrada, como se mencionó anteriormente.

+ info

En la franja de color naranja podemos considerar que la complejidad algorítmica es aceptable en determinados casos. A partir de la notación O(n log n) en la franja roja, los algoritmos empiezan a consumir mayor cantidad de recursos computacionales y tiempo dependiendo de la entrada que reciban, lo que los vuelve ineficientes, indicando que el proceso se podría optimizar de alguna forma.

Ahora que se conocen los conceptos de complejidad algorítmica y la notación Big O para categorizarla, es tiempo de ver cómo afectan al blockchain.

La complejidad algorítmica es fundamental en el contexto de blockchain, ya que afecta directamente el rendimiento y la eficiencia de las operaciones en la red. El blockchain, como estructura descentralizada y distribuida, requiere la ejecución de diversos algoritmos para realizar tareas como consenso, validación de transacciones y seguridad criptográfica.

Estas son algunas de las formas en las que la complejidad algorítmica puede afectar al blockchain:

- **Velocidad de Consenso**
- **Escalabilidad**
- **Validación de Transacciones**
- **Eficiencia Energética**
- **Seguridad Criptoeconómica**

Eficiencia Energética

Algoritmos de minería menos intensivos computacionalmente pueden reducir el consumo de energía de la red. La elección de algoritmos en el diseño de blockchain es crucial para equilibrar el consumo, la seguridad, la velocidad y la escalabilidad.

Velocidad de Consenso

En sistemas blockchain que utilizan algoritmos de consenso, como Prueba de Trabajo (PoW), la complejidad algorítmica afecta la velocidad con la que los nodos pueden llegar a un acuerdo sobre el estado de la cadena de bloques. Algoritmos de consenso eficientes, con menor complejidad, permiten alcanzar consensos más rápidos, lo que mejora la velocidad de confirmación de las transacciones.

Seguridad Criptoeconómica

En blockchains como Ethereum, donde la seguridad se basa en principios criptoeconómicos, la complejidad algorítmica de los contratos inteligentes y la ejecución de código tienen un papel crucial. Algoritmos criptográficos seguros son esenciales para garantizar la integridad y la inmutabilidad de los datos almacenados en el blockchain.

Validación de Transacciones

También influye en la velocidad y eficiencia con la que se validan las transacciones en la red. Algoritmos de validación más eficientes contribuyen a una mayor capacidad de procesamiento de transacciones por unidad de tiempo. La elección de algoritmos de firma digital, hash y criptográficos utilizados para verificar las transacciones impacta directamente en la complejidad y el rendimiento.

Escalabilidad

La complejidad algorítmica puede tener un impacto significativo en la escalabilidad de la red blockchain. Algoritmos más eficientes permiten manejar un mayor volumen de transacciones y un crecimiento sostenible de la red.