

Comenzado el lunes, 27 de marzo de 2023, 06:08**Estado** Finalizado**Finalizado en** lunes, 27 de marzo de 2023, 07:05**Tiempo empleado** 57 minutos 7 segundos**Calificación** Sin calificar aún**Pregunta 1**

Finalizado

Puntúa como 1.00

¿Por qué es importante medir la complejidad de un algoritmo? ¿Cuándo se hace requerida su validación? Argumente su respuesta y complemente con ejemplos.

Es importante medir la complejidad de un algoritmo porque nos permite evaluar su eficiencia y determinar cuánto tiempo y espacio se requieren para realizar operaciones en la estructura de datos.

La elección adecuada de la estructura de datos y el algoritmo es clave para optimizar tanto el tiempo como el espacio requerido al resolver un problema computacional.

Si se elige una estructura de datos o un algoritmo ineficiente, puede resultar en una aplicación lenta o incluso inutilizable para problemas grandes.

La validación de la complejidad de un algoritmo se hace requerida cuando se necesita resolver un problema computacional que involucra grandes cantidades de datos o cuando se necesita optimizar el rendimiento de una aplicación.

Por ejemplo:

1. Si se necesita realizar una gran cantidad de búsquedas en una estructura de datos, es importante elegir una estructura de datos que permita búsquedas eficientes, como un árbol de búsqueda binario.
2. "Si se necesita realizar muchas inserciones y eliminaciones en la estructura de datos, es importante elegir una estructura de datos que permita inserciones y eliminaciones eficientes, como una lista enlazada".

Además, la validación de la complejidad de un algoritmo es importante en la planificación y diseño de sistemas informáticos y algoritmos. Por ejemplo, si se está diseñando un sistema de búsqueda de información en una base de datos grande, es importante elegir un algoritmo de búsqueda eficiente que tenga una complejidad de $O(\log n)$ en lugar de un algoritmo de búsqueda ineficiente que tenga una complejidad de $O(n)$.

En resumen, medir la complejidad de un algoritmo es importante para evaluar su eficiencia y optimizar el rendimiento de una aplicación. La validación de la complejidad de un algoritmo se hace requerida cuando se necesita resolver un problema computacional que involucra grandes cantidades de datos o cuando se necesita optimizar el rendimiento de una aplicación.

Pregunta 2

Finalizado

Puntúa como 1.00

Describa por medio de un ejemplo en qué consiste la notación Big O

La notación Big O es una técnica utilizada en el análisis algorítmico para describir la tasa de crecimiento del tiempo o del espacio requerido a medida que el tamaño del problema aumenta en las estructuras de datos. Por ejemplo, si tenemos un algoritmo que realiza una operación en una lista de tamaño n , y el tiempo que tarda en realizar la operación es proporcional a n^2 , entonces decimos que la complejidad del algoritmo es $O(n^2)$. Un ejemplo concreto de la notación Big O sería el algoritmo de ordenamiento de burbuja. Este algoritmo consiste en comparar pares de elementos adyacentes en una lista y cambiarlos de posición si están en el orden incorrecto. "Este proceso se repite varias veces hasta que la lista esté ordenada". La complejidad de este algoritmo es $O(n^2)$, ya que en el peor de los casos, se deben realizar n^2 comparaciones y n^2 intercambios de elementos. Por lo tanto, al utilizar la notación Big O, podemos evaluar la eficiencia de un algoritmo y determinar cuánto tiempo y espacio se requieren para realizar operaciones en la estructura de datos. Esto nos permite elegir la estructura de datos y el algoritmo adecuados para optimizar el rendimiento de una aplicación.

Pregunta **3**

Finalizado

Puntúa como 1.00

¿Cómo podría influir la notación Big O en técnicas más avanzadas como Big Data? Argumente su respuesta

La notación Big O es una técnica fundamental en el análisis algorítmico y es esencial para la optimización de algoritmos y estructuras de datos.

En el contexto de Big Data, la notación Big O sigue siendo importante porque el análisis de grandes conjuntos de datos requiere algoritmos y estructuras de datos eficientes para manejar la gran cantidad de información.

En Big Data, la cantidad de datos que se manejan es enorme y, por lo tanto, es importante tener en cuenta la complejidad de los algoritmos y la eficiencia de las estructuras de datos para manejar estos datos de manera eficiente.

La elección adecuada de la estructura de datos y el algoritmo es clave para optimizar tanto el tiempo como el espacio requerido al procesar grandes conjuntos de datos.

Por ejemplo, si se necesita realizar una búsqueda en un conjunto de datos masivo, es importante elegir un algoritmo de búsqueda eficiente que tenga una complejidad de $O(\log n)$ en lugar de un algoritmo de búsqueda ineficiente que tenga una complejidad de $O(n)$. De esta manera, se puede reducir significativamente el tiempo requerido para realizar la búsqueda.

Además, la notación Big O también es importante en Big Data porque los recursos de hardware son limitados.

Al elegir algoritmos y estructuras de datos eficientes, se puede reducir la cantidad de recursos de hardware necesarios para procesar grandes conjuntos de datos. Esto puede resultar en un ahorro significativo de costos y una mayor eficiencia en el procesamiento de grandes conjuntos de datos.

En resumen, la notación Big O sigue siendo importante en técnicas más avanzadas como Big Data porque el análisis de grandes conjuntos de datos requiere algoritmos y estructuras de datos eficientes para manejar la gran cantidad de información. La elección adecuada de la estructura de datos y el algoritmo es clave para optimizar tanto el tiempo como el espacio requerido al procesar grandes conjuntos de datos y reducir la cantidad de recursos de hardware necesarios.

Pregunta **4**

Finalizado

Puntúa como 1.00

Describa con sus palabras y argumente con un ejemplo la función de un algoritmo

Un algoritmo es un conjunto de instrucciones que se utilizan para resolver un problema o realizar una tarea específica.

La función de un algoritmo es proporcionar una solución paso a paso para un problema determinado.

Los algoritmos se utilizan en una amplia variedad de campos, no solo en la ingeniería de sistemas.

Por ejemplo, un algoritmo para sumar dos números enteros, pasos:

1. Pedir al usuario que ingrese el primer número.
2. Leer y almacenar el primer número ingresado por el usuario en una variable.
3. Pedir al usuario que ingrese el segundo número.
4. Leer y almacenar el segundo número ingresado por el usuario en otra variable.
5. Sumar los dos números ingresados y almacenar el resultado en una tercera variable.
6. Mostrar el resultado de la suma al usuario.

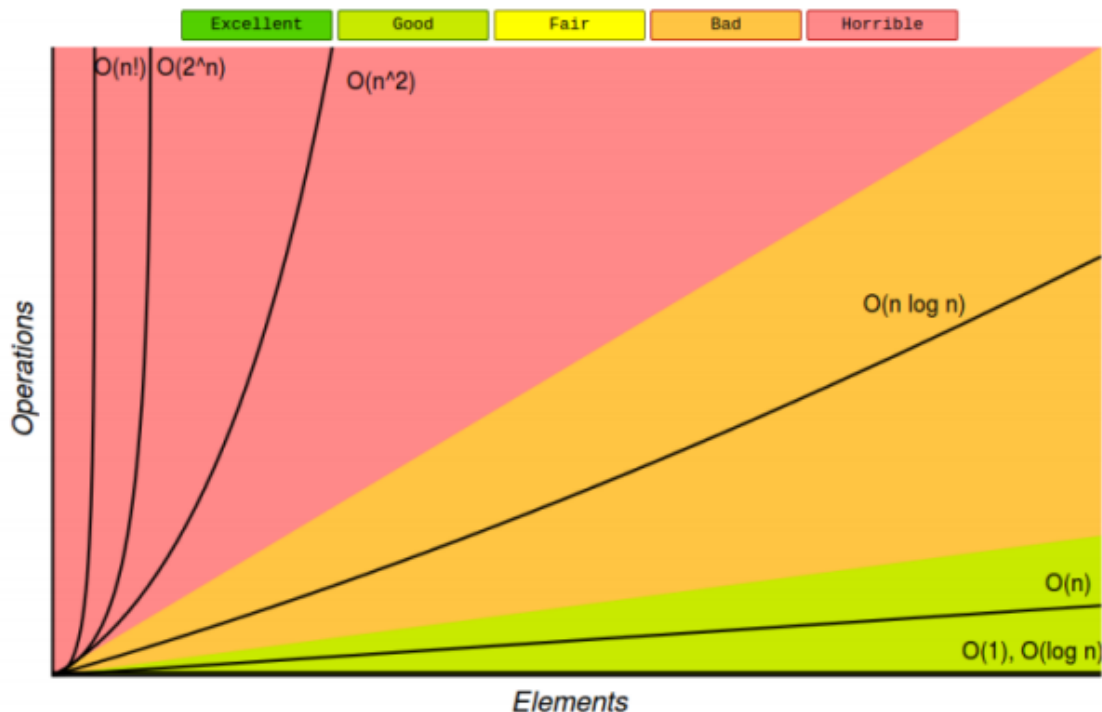
En resumen, la función de un algoritmo es proporcionar una solución paso a paso para un problema determinado. Los algoritmos se utilizan en una amplia variedad de campos y son esenciales para la resolución de problemas y la realización de tareas específicas, desde problemas básicos como la suma de dos números, hasta problemas muy complejos como lo es la creación de una red neuronal para machine learning de aprendizaje no supervisado con tensorflow.

Pregunta **5**

Finalizado

Puntúa como 1.00

Describe utilizando un ejemplo, el significado y representación de la siguiente gráfica



La gráfica de Big O es una representación visual de la complejidad de un algoritmo en función del tamaño del problema. En esta gráfica, se representa el tamaño del problema y el color es la "calificación" de espacio y tiempo requerido para resolver el problema.

La gráfica de Big O muestra cómo el tiempo o el espacio requerido aumenta a medida que el tamaño del problema aumenta.

Explicación de las líneas: si tenemos un algoritmo con una complejidad de $O(n)$, la gráfica de Big O mostrará una línea recta que se extiende desde el origen y se mueve hacia arriba a medida que el tamaño del problema aumenta. Esto significa que el tiempo o el espacio requerido aumenta linealmente con el tamaño del problema. Por otro lado, si tenemos un algoritmo con una complejidad de $O(n^2)$, la gráfica de Big O mostrará una curva que se extiende hacia arriba a medida que el tamaño del problema aumenta. Esto significa que el tiempo o el espacio requerido aumenta exponencialmente con el tamaño del problema. La gráfica de Big O es una herramienta útil para evaluar la eficiencia de un algoritmo y determinar cuánto tiempo y espacio se requieren para resolver un problema en función del tamaño del problema. Al utilizar la gráfica de Big O, podemos comparar diferentes algoritmos y elegir el más eficiente para resolver un problema determinado.

Ejemplo: Buscar un número en una lista, este sería del tipo $O(n)$, es decir, que su tiempo de ejecución aumenta linealmente con el tamaño de la entrada (en este caso, la cantidad de elementos en la lista). Si la lista es muy grande, la función podría tardar mucho tiempo en ejecutarse, como se puede observar, de acuerdo a la gráfica dada, si corresponde a un comportamiento lineal que depende de la cantidad n de datos almacenados en la lista

código:

◀ Notas de Clase: Introducción a la complejidad de algoritmos

Volver a: Análisis Algorí... ➡