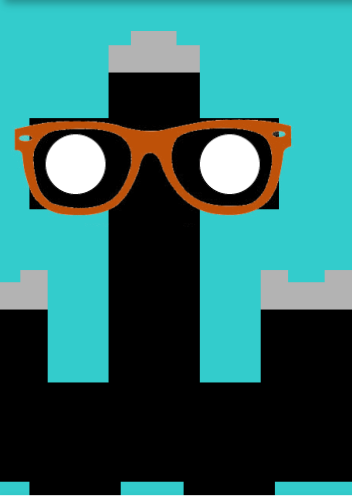




Introducción

Pilas Árboles List
Colas
Stack Tree
Array
Queue Listas

¿Son importantes?

Las estructuras de datos definen la forma en que los datos se organizan en la memoria para el acceso rápido y eficiente, mientras que los algoritmos son un conjunto de instrucciones para resolver problemas al manipular estas estructuras de datos.

Diseñar un algoritmo eficiente es una habilidad muy importante que toda empresa de software persigue, por ejemplo Microsoft, Google, Facebook, etc.

La mayoría de las entrevistas para estas empresas se centran en el conocimiento de estructuras de datos y algoritmos. Buscan que los candidatos usen conceptos de estructuras de datos y algoritmos para resolver problemas complejos de manera eficiente. Aparte, saber un lenguaje de programación, ya que también se necesita para tener un buen dominio de estos fundamentos informáticos no solo para calificar en la entrevista, sino también para sobresalir en sus trabajos como ingeniero de software.



Introducción

Aprendemos por experiencia. Con experiencia, es fácil resolver nuevos problemas. Al analizar varios algoritmos o técnicas de resolución de problemas, comenzamos a desarrollar un patrón que nos ayudará a resolver problemas similares.

Un algoritmo es un conjunto de pasos para realizar una tarea. Un algoritmo es un programa de computadora, es un conjunto de pasos que se aplican sobre un conjunto de entrada para producir un conjunto de resultados.

El conocimiento del algoritmo nos ayuda a obtener el resultado deseado más rápido aplicando el algoritmo apropiado.

Las propiedades más importantes de un algoritmo son:

1. **Corrección:** el algoritmo debe ser correcto. Debería poder procesar todas las entradas dadas y proporcionar una salida correcta.
2. **Eficiencia:** el algoritmo debe ser eficiente en la resolución de problemas. La eficiencia se mide en dos parámetros. El primero es la complejidad del tiempo, la rapidez con que un algoritmo proporciona el resultado. En segundo lugar está la complejidad espacial, qué es cuánta RAM o memoria va a consumir un algoritmo para dar el resultado deseado.

El tiempo de complejidad espacial está representada por la función $T(n)$ - tiempo requerido versus tamaño de entrada n .

La complejidad espacial está representada por la función $S(n)$ - memoria usada versus el tamaño de entrada n .