
PIXEL PRINT STUDIO

FASE 3

202201947– PABLO ANDRES RODRIGUEZ LIMA

Resumen

Los árboles en estructuras de datos son un tipo de estructura jerárquica que consta de nodos interconectados. Cada nodo tiene un valor y puede tener cero o más nodos hijos, organizados en una estructura de árbol. Estos árboles se utilizan para representar relaciones jerárquicas entre elementos, como estructuras de archivos en sistemas operativos o relaciones de parentesco en genealogías. Ofrecen eficiencia en la búsqueda, inserción y eliminación de elementos, siendo esenciales en la implementación de algoritmos y aplicaciones informáticas complejas.

Abstract

Trees in data structures are hierarchical structures consisting of interconnected nodes. Each node holds a value and can have zero or more child nodes, organized in a tree-like structure. These trees are used to represent hierarchical relationships between elements, such as file structures in operating systems or parent-child relationships in genealogies. They offer efficiency in searching, inserting, and deleting elements, being essential in implementing complex algorithms and computer applications.

Palabras clave

1. Jerárquico
2. Nodos
3. Relaciones padre-hijo
4. Búsqueda
5. Eficiencia

Keywords

1. Hierarchical
2. Nodes
3. Parent-child Relationships
4. Search
5. Efficiency

Introducción

En el mundo de la programación, las Estructuras de Datos Abstractas (EDA) son fundamentales para organizar y manipular datos de manera eficiente. Una EDA es una colección de datos y un conjunto de operaciones definidas en esos datos, independientes de cualquier implementación concreta. Estas estructuras proporcionan un marco conceptual para entender cómo los datos están organizados y cómo se puede acceder a ellos y manipularlos.

Desarrollo del tema

1. Tabla Hash:

Una tabla hash es una estructura de datos que se utiliza para almacenar y recuperar datos de manera eficiente, donde los datos se indexan mediante una función hash.

Características:

Utiliza una función hash para calcular la ubicación de almacenamiento de los datos en la tabla.

Permite acceso rápido a los datos, con un tiempo de acceso promedio constante ($O(1)$) en el mejor caso.

Es especialmente útil para la implementación de diccionarios, mapas o conjuntos en lenguajes de programación.

La eficiencia de la tabla hash depende de la calidad de la función hash y de la resolución de colisiones

2. Algoritmo de Dijkstra:

El algoritmo de Dijkstra es un algoritmo de búsqueda de caminos más cortos en un grafo ponderado, utilizado para encontrar la ruta más corta entre un nodo de inicio y todos los demás nodos en el grafo.

Características:

Utiliza un enfoque de búsqueda primero el mejor (best-first search), seleccionando en cada paso el nodo con el costo más bajo.

Funciona óptimamente en grafos dirigidos y no dirigidos con pesos no negativos.

Es eficiente para encontrar rutas cortas en redes de comunicación y sistemas de transporte.

3. Árbol de Merkle:

Un árbol de Merkle es una estructura de datos en forma de árbol binario donde cada hoja está etiquetada con el hash de un bloque de datos, utilizado para garantizar la integridad de los datos.

Características:

Permite verificar rápidamente la integridad de grandes conjuntos de datos sin tener que comparar los datos completos.

Se utiliza en sistemas de almacenamiento distribuido y criptomonedas para asegurar la integridad de los datos.

Los cambios en los datos requieren la actualización de solo unas pocas porciones del árbol, lo que lo hace eficiente para grandes conjuntos de datos.

4. Blockchain:

Blockchain es una tecnología de contabilidad distribuida que proporciona un registro inmutable y transparente de transacciones, construido sobre una cadena de bloques enlazados.

Características:

Proporciona un registro inmutable de transacciones confirmadas, donde cada bloque contiene un hash que apunta al bloque anterior.

Es descentralizado y seguro, ya que los datos están distribuidos en múltiples nodos de la red y están protegidos mediante criptografía.

Se utiliza en aplicaciones como criptomonedas, contratos inteligentes y seguimiento de la cadena de suministro.

5. Grafos:

Un grafo es una estructura que consiste en un conjunto de nodos (vértices) conectados por aristas (bordes), utilizada para modelar relaciones entre objetos.

Características:

Los grafos pueden ser dirigidos o no dirigidos, dependiendo de si las aristas tienen dirección.

Pueden ser ponderados o no ponderados, dependiendo de si las aristas tienen peso asociado.

Se utilizan para modelar una variedad de situaciones, como redes sociales, sistemas de transporte y relaciones de dependencia.

Existen numerosos algoritmos asociados con grafos para resolver problemas como búsqueda de caminos más cortos, recorridos y flujo máximo.

Árbol AVL: Un árbol AVL es un tipo de árbol binario auto-balanceado, lo que significa que la altura de cada subárbol siempre es similar. Esta propiedad permite realizar operaciones de búsqueda, inserción y eliminación de forma eficiente.

Los árboles AVL son más eficientes que los árboles binarios normales para operaciones de búsqueda, inserción y eliminación.

Se utilizan en aplicaciones donde la velocidad de acceso a los datos es importante.

FORTTRAN FPM para la Lectura de JSON: FPM (Flexible Parsing Model) es una extensión del lenguaje de programación FORTRAN que permite leer y escribir archivos JSON de forma sencilla. FPM proporciona una serie de herramientas para analizar la estructura del archivo JSON y acceder a sus datos.

- FPM es una herramienta poderosa para leer y escribir archivos JSON en FORTRAN.
- FPM facilita el acceso a los datos dentro del archivo JSON.

Conclusiones

1. La combinación de tablas hash, el algoritmo de Dijkstra y los árboles de Merkle ha demostrado

ser una estrategia efectiva para garantizar la eficiencia en el acceso a los datos, la optimización de rutas y la seguridad de la integridad de los datos en nuestro proyecto.

2. La incorporación de la tecnología blockchain ha fortalecido la seguridad y transparencia de nuestro sistema, proporcionando un registro inmutable y confiable de transacciones, así como la capacidad de ejecutar contratos inteligentes de manera segura y transparente.
3. El modelado de relaciones con grafos ha sido fundamental para representar las interacciones entre entidades en nuestro sistema, permitiendo una visualización clara y precisa de las conexiones y optimizando la toma de decisiones informadas en diferentes contextos.

Referencias bibliográficas

Programación II - Tipo de Dato Abstracto. (n.d.).

<https://sites.google.com/site/programacioniiu/no/temario/unidad-2---tipo-abstrcto-de-dato/tipo-de-dato-abstrcto>

Estructuras de datos: listas enlazadas, pilas y colas.

(n.d.).

<https://calcifer.org/documentos/librognome/glib-lists-queues.html>