Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2024

Listas enlazadas y su aplicación en Python: camio y volteo de azulejos

202200220 – Maria Paola Guadalupe Dávila Valenzuela

Resumen

El desarrollo del programa incluye la implementación de listas vinculadas para administrar plantillas cargadas desde archivos XML. Los usuarios pueden buscar patrones similares en la lista relacionada especificando el nivel, el código de inicio y el código de finalización. El programa calcula el costo mínimo para convertir a la muestra final deseada y muestra el proceso de conversión junto con el costo total. Además, los usuarios pueden ver información detallada sobre los nodos y modelos disponibles, incluidos tamaños, costos asociados y código fuente relacionado. Esta aplicación es una herramienta eficaz para gestionar y buscar patrones en diferentes aplicaciones.

Palabras clave

Listas enlazadas, Estructura de datos, Nodos, Implementación de Python, ventajas y desventajas, aplicaciones prácticas, optimización de rendimiento, gestión de memoria, complejidad algorítmica.

Abstract

The program development involves the implementation of linked lists to manage templates loaded from XML files. Users can search for similar patterns in the linked list by specifying the level, start code, and end code. The program calculates the minimum cost to convert to the desired final sample and displays the conversion process along with the total cost. Additionally, users can view detailed information about available nodes and models, including sizes, associated costs, and related source code. This application is an effective tool for managing and searching for patterns in various applications.

Keywords

Linked lists, Data structure, Nodes, Python implementation, Advantages and disadvantages, Practical applications, Performance optimization, Memory management, Algorithmic complexity.

Introducción

Las listas enlazadas son datos estructurados fundamentales en la informática y programación. Tienen la capacidad de almacenar y organizar datos de forma dinámica, lo que las vuelve fundamentales en la implementación de algoritmos y la gestión de memoria.

Las listas enlazadas se basan en el concepto de nodos interconectados, donde cada nodo guarda un elemento en su interior y una referencia al siguiente nodo en su secuencia, esto ofrece una alternativa a las listas estáticas o tradicionales ya que las listas enlazadas no requieren de una asignación de memoria continua y se pueden adaptar a dinámicamente a los cambios dentro del tamaño de los datos.

Este ensayo tiene como propósito explorar la importante y trascendencia de las listas enlazadas dentro de la programación en Python.

Desarrollo del tema

Concepto y estructura de lista enlazada.

Las listas enlazadas son estructuras de datos fundamentales en informática y programación. Almacenan datos de manera organizada en nodos, donde cada nodo contiene un valor y un enlace al siguiente nodo en secuencia. A diferencia de una matriz donde los elementos se almacenan en ubicaciones de memoria contiguas.

En la estructura básica de una lista vinculada, los nodos contienen dos partes principales: datos para almacenar información y un puntero en la lista vinculada que apunta al siguiente nodo. El último nodo de la lista enlazada no apunta a ningún otro nodo, lo que indica el final de la lista enlazada.

Esta estructura permite la adición y eliminación eficiente de elementos porque no requiere una reorganización constante de elementos como en una matriz. Sin embargo, acceder a los elementos de forma discreta puede ser menos eficiente porque hay que recorrer la lista desde el principio o desde una ubicación conocida.

Ventajas y desventajas de las listas enlazadas

Las listas enlazadas ofrecen muchos beneficios, incluida la capacidad de agregar y eliminar elementos en cualquier posición con complejidad O(1), es decir, en tiempo constante. Esto los hace ideales para aplicaciones que requieren una gestión rápida de datos. Sin embargo, las listas enlazadas también tienen algunas limitaciones. Tienen un acceso aleatorio menos eficiente que las matrices porque acceder a un elemento requiere atravesar la lista desde el principio o desde un punto conocido, lo que puede conducir a una complejidad O(n) en el peor de los casos. Además, almacenar punteros adicionales puede aumentar el uso de memoria en comparación con las matrices.

Aplicaciones prácticas y aplicaciones en Python.

La lista incluye muchas aplicaciones de programación general y Python. Se puede utilizar en proyectos con varias combinaciones, por ejemplo, baterías, coque y plantas. Además, son útiles en situaciones que requieren una gestión eficiente de la memoria y una alta flexibilidad en la manipulación de datos.

Por ejemplo, las listas enlazadas se pueden leer e implementar en implementaciones algorítmicas y de ordenación, como encadenamiento u orden de inserción. También son útiles cuando se gestionan grandes conjuntos de datos que requieren la inserción y eliminación frecuentes de elementos.

Implementación de la lista vinculada de Python

Las listas enlazadas se pueden implementar en Python de muchas maneras. Una implementación típica es crear una clase Node para representar nodos individuales y una clase LinkedList para representar la lista enlazada completa. Esta implementación permite un manejo flexible de la lista y le permite agregar fácilmente funcionalidades adicionales, como métodos para buscar elementos, eliminar nodos o invertir listas.

Otra opción es una lista doblemente enlazada, donde cada nodo contiene enlaces tanto al nodo anterior como al siguiente de la cadena. Esto permite el acceso bidireccional a los elementos de la lista, lo que puede resultar útil en algunas situaciones.

Rendimiento optimizado y gestión de memoria

En aplicaciones donde el rendimiento y la eficiencia de la memoria son críticos, es importante optimizar las implementaciones de listas vinculadas. Esto puede incluir el uso de técnicas como la asignación y liberación de memoria de manera eficiente, el uso de nodos preasignados en lugar de la asignación dinámica y el uso de iteradores para particionar la lista de manera eficiente y efectiva.

También es importante considerar el manejo de casos especiales, como la administración de memoria cuando las listas se eliminan o se eliminan por completo, y la implementación de mecanismos para evitar pérdidas de memoria o fragmentación excesiva.

Complejidad algorítmica y pensamiento creativo

Al diseñar algoritmos y estructuras de datos utilizando listas enlazadas, se debe tener en cuenta la complejidad física y espacial de la operación. Si bien las operaciones de inserción y eliminación siempre se pueden realizar de la mejor manera posible, es importante comprender que estas operaciones pueden resultar muy costosas si se realizan con frecuencia en condiciones normales de lista.

Por otro lado, acceder a los elementos de una lista es mucho más sencillo, especialmente cuando se utiliza un iterador para moverse por la lista en orden. Sin embargo, es importante recordar que implementar la insensibilidad al peor de los casos puede limitar la efectividad de algunos algoritmos y métodos.

Programas y extensiones

Además de las funciones básicas de inserción, eliminación y acceso, las listas vinculadas se pueden ampliar con funciones adicionales según los requisitos de la aplicación. Por ejemplo, puede implementar métodos para buscar elementos, ordenar listas, crear sumas o asignaciones a listas y más. De manera similar, las listas vinculadas se pueden utilizar como base para implementar estructuras de datos complejas, como listas circulares, listas doblemente vinculadas, matrices, colas, árboles e imágenes. Estas innovaciones nos permiten resolver muchos problemas y aplicaciones en informática y programación.

En el caso de inclusión de figuras, deben ser nítidas, legibles en blanco y negro. Se denomina figuras a gráficas, esquemas, fotografías u otros elementos gráficos.

Desarrollo del programa

Editor de plantillas: el programa te permite cargar plantillas desde archivos XML, donde cada plantilla está asociada a una categoría específica. Estas

funciones se conservarán en la lista de contactos en versiones futuras.
Búsqueda de patrones: el usuario puede seleccionar el nivel, el código de inicio y el código de finalización. El programa busca modelos similares en la lista vinculada y calcula el costo mínimo de pasar al último modelo de muestra en la lista. Luego muestra el proceso de cambio y el costo total.
Ver información: El programa le permite ver información sobre botones y modelos disponibles, sus tamaños, costos asociados y códigos fuente correspondientes.

Conclusiones

- La implementación eficiente de algoritmos de gestión de listas enlazadas puede conducir a una optimización significativa de los recursos computacionales. Minimizar el tiempo de ejecución y el consumo de memoria mejora el rendimiento general del programa. Comprender los algoritmos de inserción, eliminación y búsqueda de listas vinculadas es importante para maximizar el rendimiento y minimizar los cuellos de botella de las aplicaciones.

- Las listas enlazadas son una estructura de datos extensible y fácil de mantener. El modularidad inherente de las listas vinculadas facilita agregar nuevas funciones y cambiar las existentes sin afectar otras partes del código. Esta característica es especialmente valiosa en grandes proyectos de desarrollo de software donde la capacidad de adaptarse rápidamente a los cambios es fundamental.

- Las listas enlazadas se utilizan en muchos campos diferentes, desde informática hasta ingeniería, matemáticas y más. Se utilizan para implementar estructuras de datos más complejas, como colas, pilas, árboles, y para resolver problemas en áreas como inteligencia artificial, bioinformática y visualización de datos.

- Dominar las listas enlazadas es importante para todos los programadores, tanto principiantes como experimentados. Comprender los conceptos básicos y practicar su implementación son habilidades valiosas que se pueden aplicar para resolver muchos problemas de programación. Además, aprender listas enlazadas promueve el pensamiento lógico y la resolución de problemas, que son habilidades esenciales en informática y más allá.

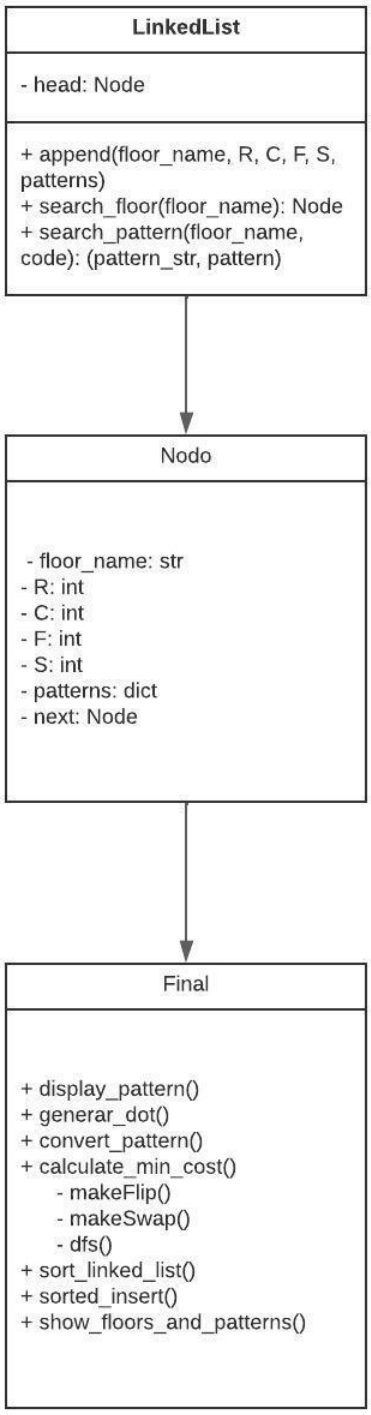
Referencias bibliográficas

Archivos de documentación Python 3.12.2
https://docs.python.org/3/

GitHub, Repositorio Auxiliar Andrea Cabrera:
https://github.com/AndreaCabrera01/1S2024_IPC2

Listas enlazadas de Python
https://pythondiario.com/2018/07/linked-list-listas-enlazadas.html

Anexos


LinkedList
- head: Node
+ append(floor_name, R, C, F, S, patterns)
+ search_floor(floor_name): Node
+ search_pattern(floor_name, code): (pattern_str, pattern)
Nodo
- floor_name: str
- R: int
- C: int
- F: int
- S: int
- patterns: dict
- next: Node
Final
+ display_pattern()
+ generar_dot()
+ convert_pattern()
+ calculate_min_cost()
- makeFlip()
- makeSwap()
- dfs()
+ sort_linked_list()
+ sorted_insert()
+ show_floors_and_patterns()