

PROYECTO 1- IPC2-SECCIÓN D

Resumen

Los TDA son como dicen sus siglas un tipo de dato abstracto, se le conoce también como un modelo matemático por su contenido que es un conjunto de operaciones a realizar para un grupo de datos. Existen cuatro tipos de TDA que son: TDA lista, TDA pila, TDA cola y TDA cola de prioridad. Un TDA va de la mano con lo que son las listas enlazadas, ya sean simples, dobles, circulares o inclusive doblemente circulares para tener una referencia del nodo anterior, el siguiente y que el último nodo apunte al primero para lograr una mayor facilidad en las operaciones a efectuar en cada nodo. La lista más sencilla de implementar ya que cada nodo apunta al siguiente y el último apunta a null, una de las más complejas a utilizar es la circular doblemente enlazada por las referencias que puede hacer cada nodo, el llevar su secuencia se puede complicar dependiendo la instrucción.

Palabras clave

1. TDA
2. Lista Circular
3. FIFO
4. Lista Doblemente Enlazada
5. Nodo

201905152 – Brian Antonio Hernandez Gil

Abstract

TDA's are, as an abstract data type say, it is also known as a mathematical model for its content that is a set of operations to perform for a group of data. There are four types of TDA that are: TDA List, Stack TDA, TDA Queue, and TDA Priority Queue. A TDA goes hand in hand with what are linked lists, whether single, double, circular, or even double-circular to have a reference from the previous node, the next node, and for the last node to point to the first node for easier operations on each node. The easiest list to implement because each node points to the next and the last one points to null, one of the most complex to use is the circular double bound by the references that each node can make, carrying its sequence can be complicated depending on the statement.

Keywords

1. TDA
2. Circular list
3. FIFO
4. Doubly Linked List
5. Node

Introducción

Lo que es la implementación de TDA de cualquier tipo en un programa es funcional porque nos permite almacenar para después poder realizar operaciones de la lista de diferentes formas en cada nodo dependiendo como lo desee el cliente que se establezca en el programa. El uso de TDA tiene una gran ventaja que es el producir código reutilizable lo cual ayuda mucho al programador, la realización de este proyecto lleva TDA y listas enlazadas para la recolección de los datos de la matriz, lo cual facilita la separación de los datos en diferentes listas simples o dobles, todo dependiendo del uso que se le vaya a brindar a esos datos almacenados. Para la resolución de este enunciado el fin es el almacenamiento de las variables localizadas dentro del archivo de extensión xml lo cual nos involucra una y, x, nombre, n, m y el dato.

Desarrollo del tema

El proyecto trata de la implementación de TDA y listas enlazadas para con ello poder almacenar los datos, convertirlos a binarios y luego compararlos para encontrar coincidencias y sumar sus valores verdaderos, la parte de graficar que se realizó con la librería Graphviz utilizando los `dot.node()` y mandando a llamar los atributos que poseía cada matriz por medio de su Key y utilizando un `for` para ya con ello ir agregando los datos de forma dinámica y que no fueran estáticos porque si lo fueran en lo que es los datos o bueno más bien los elementos de la matriz imprimiría en la imagen de extensión png el último elemento que posee dicha matriz en lugar de todos. El

TDA como se explicó en el resumen y brevemente en la introducción es un tipo de dato abstracto y prácticamente una estructura donde se van a ir almacenando los datos de manera sucesiva para ya con ello mandarlos a llamar, para almacenarlos y retirarlos existen dos formas las cuales es la FIFO y LIFO. La primera mencionada que es la FIFO explica que el primer dato que entra a la lista es el primero en ser extraído de la misma, ahora la segunda explica que el último dato en entrar a la lista es el primero en salir de la misma.

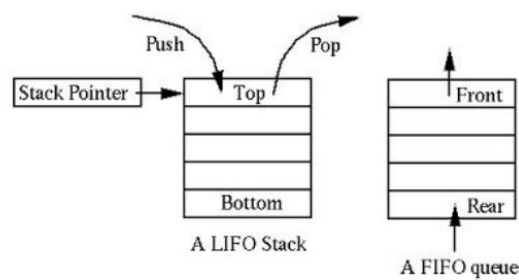


Figura 1. Imagen de FIFO y LIFO
Fuente: Curso de Gambas 3

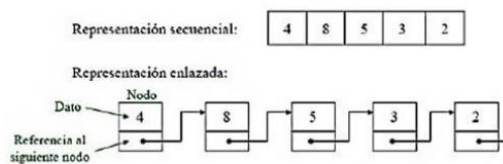


Figura 2. Imagen Lista Enlazada
Fuente: Monografias.com

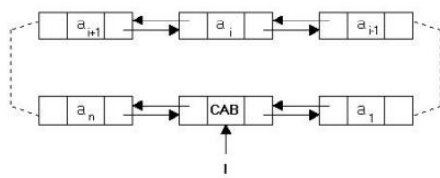


Figura 3. Imagen Lista Doblemente Enlazada
Fuente: Curso de Gambas 3

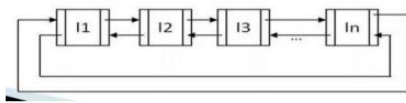


Figura 4. Imagen Lista Doblemente Circular
Fuente: Curso Gambas 3

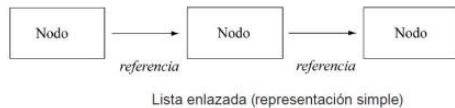


Figura 5. Imagen de Nodos en una lista simple
Fuente: Curso Gambas 3

Como se observa anteriormente se colocaron varias ilustraciones para que usted el lector pueda observar y analizar cómo son los TDA, las listas y los nodos que es lo que posee cada una de ellas.

En la opción uno del menú del programa se encuentra la opción de cargar archivo, la misma añadirá los valores o más bien los elementos de la matriz a una lista simple en donde pueden ser llamados cuando se utilicen, además a ello imprime los elementos de esa dicha matriz, en la opción numero dos hará la suma de tuplas y demás, prácticamente resolverá la matriz reducida y estará lista para ser mostrada después. En la opción 3 se ingresará el nombre del archivo de entrada que es donde se guardará el xml de la matriz reducida. En la opción cuatro el programa imprimirá lo que son datos personales de la persona que realizó el programa, la opción 5 está el algoritmo el cual despliega un grafo con la matriz, su nombre y sus datos, estos últimos son el número de filas y el número de columnas que posee. Este algoritmo lo que hace es

pedirle al usuario el nombre de la matriz que desea graficar, para ya con ello buscarla y con el recorrido que hay dentro de ella obtener los datos que necesita para ser desplegados en un archivo de extensión png, para el apartado de graficar la matriz tal como es, se realizó un algoritmo que consiste en un for que recorre el archivo para extraer sus datos y dentro de él tres if anidados los cuales ayudarán al for a que imprima los datos de manera dinámica por medio de un contador de filas y otro de columnas concatenando los datos y generando nuevos `dot.node()` de manera dinámica para ya con ello generar las filas, columnas e imprimir los elementos de manera clara y ordenada hasta que ya no exista ninguno dentro del archivo de extensión xml y en el punto 6 uno elige salir del programa.

Conclusiones

Después de la realización de este proyecto concluyó que los TDA son importante para el manejo de datos, abstracción de estos y operaciones a realizar con cada uno de ellos porque a veces hay que hacer muchas operaciones con esos datos y almacenarlos en una lista enlazada tenemos la ventaja de poder reutilizar código mismo de ello para otros métodos

Conclusiones Después de la realización de este proyecto concluyó que los TDA son importante para el manejo de datos, abstracción de estos y operaciones a realizar con cada uno de ellos porque a veces hay que hacer muchas operaciones con esos datos y almacenarlos en una lista enlazada tenemos la ventaja de poder reutilizar código mismo de ello para otros métodos propenso a cometer errores al igual que no se puede reutilizar código.

El manejo de la librería Graphviz es esencial para exportación de imágenes, documentos y muchas cosas más ya que esta librería permite al programador tener muchos caminos para realizar las operaciones y demasiadas opciones para decorar o agregar detalles importantes a un documento o una imagen, al ser curioso me percate de que posee una gran variedad para el trabajo con el colores, ya sea colores normales y solidos como degradados lo cual a mi parecer es algo perfecto ya que le da al programador como mencione anteriormente muchas opciones para que pueda hacer lo que le soliciten y lo que su imaginación cree para dicho programa.

Referencias bibliográficas

- Antonakos, James L. and Mansfield, Kenneth C., Jr. Practical Data Structures Using C/C++ (1999). Prentice-Hall. ISBN 0-13-280843-9, pp. 165–190
- Collins, William J. Data Structures and the Java Collections Framework (2002,2005) New York, NY: McGraw Hill. ISBN 0-07-282379-8, pp. 239–303
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford Introductions to Algorithms (2003). MIT Press. ISBN 0-262-03293-7, pp. 205–213, 501–505
- Donald Knuth. Fundamental Algorithms, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89683-4. Sections 2.2.3–2.2.5, pp.254–298.
- .National Institute of Standards and Technology (August 16, 2004). Definition of a linked list. Retrieved December 14, 2004.