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 2022.

Proyecto 1 IPC2 TDA's

20111718 – Walter Javier Santizo Mazariegos

Resumen

Utilizando el lenguaje de programación python con el cual se desarrollara este proyecto, implementado estructuras de datos TDA'S y paradigmas de programación como programación orientada a objetos (POO),imperativa, y secuencial se desarrolló una solución integral que implementa tipos de datos abstractos, bajo el concepto de programación orientada a objetos, con lo que se partió para realizar un programa en consola capaz de leer y manipular archivos con extensión XML que almacenan información sobre determinado problema por lo que una vez leído los datos se procede a manipularlos con estructuras de datos para posteriormente brindar una solución integral a un problema especifico que es el estudio de células infectadas a un resolución general como lo es el programa que se desarrollo para analizar miles de pacientes si fuese necesario, además se brinda herramienta de reporteria que permite observar la confiabilidad y viabilidad de un proyecto de estas características con grapphviz y XML de salida

Palabras clave

Python, estructura de datos, listas, listas enlazadas, listas doblemente enlazadas, programación orientada a objetos, XML, Graphviz,

Abstract

Using the python programming language with which this project was developed, implementing TDA'S data structures and programming paradigms such as object-oriented programming (OOP), imperative, and sequential, an integral solution was developed that implements abstract data types, under the concept of object-oriented programming, with which it was started to make a console program capable of reading and manipulating files with an XML extension that store information about a certain problem, so that once the data is read, it is manipulated with data structures to later it will provide a comprehensive solution to a specific problem that is the study of infected cells at a general resolution such as the program that is developed to analyze thousands of patients if necessary, in addition a reporting tool is provided that allows observing the reliability and feasibility of a project of these characteristics with grapphviz and XML d it's out

Keywords

Python, data structure, lists, linked lists, doubly linked lists, object-oriented programming, XML, Graphviz,

Introducción

La manipulación de estructuras de datos. Nos permite la organización de los datos en una computadora para que puedan ser utilizados de una manera muy eficaz. Existen diferentes tipos de estructuras de datos que son adecuados para la utilización de diferentes tipos de aplicaciones.

En esta aplicación se busca una estructura cuya forma de almacenamiento en la memoria sea la más optima. Por eso mismo la creación de una matriz dispersa es una opción muy eficaz, dado que la matriz dispersa es ampliamente usada en la computación científica, especialmente en la optimización a gran escala. Otra área de interés en donde se pueda aplicar de manera exitosa es en la teoría de grafos, teoría de redes y los métodos numéricos.

La aplicación intenta replicar la forma en la que se estudian los patrones que mediante un proceso de abstracción por el cual se representan rejillas como listas doblemente enlazadas que permiten mediante sus operaciones poder manipular toda la información almacenada en listas además de que se estudio el modelo dom de manipulación de objetos mediante los archivos XML

Desarrollo del tema

Una matriz o arreglo bidimensional, es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo, desde el punto de vista lógico una matriz puede verse como un conjunto de elementos en fila, o filas y columnas si tuviesen dos dimensiones. Cada elemento de la estructura puede ser accedida por medio del índice o posición del elemento en la matriz.

En el caso específico de esta aplicación se optó por la utilización de una matriz que será simulada mediante listas doblemente enlazadas estas iban a tener los atributos característicos de una lista doblemente enlazada, pero con los atributos extras de la fila y columna que iban a representar la posición en la cual se iban a asignar asemejándose a las estructuras que tiene Python

Cada elemento de la estructura puede ser accedida por medio del índice o posición del elemento en la matriz.
a. Es un arreglo con dos dimensiones de datos.

b. La cantidad de columnas y filas puede o no ser la misma.

c. En computación, es generalmente representada por arreglos.

La creación de una matriz dispersa (sparce matrix) es relativamente sencilla. Lo importante es conocer los apuntadores de memoria que creamos en sus métodos. Formas de implementar:

1. Arreglos.

2. Lista enlazada.

3. Enlaces de datos

4. Listas con cabecera.

A continuación, veremos cuales son las formas que se implementa con las diferentes opciones al crear una matriz dispersa.

1. Arreglos: En los arreglos existe desperdicio de memoria. Las posiciones 0, existen. Generalmente es implementado como un arreglo de dos posiciones, también es posible la utilización de un arreglo de una sola posición, pero es necesario más de algún método de indexación. La utilización de arreglos es muy eficiente para acceder a los datos. Este concepto en Python es conocido como lista de listas y la asignación de memoria para estas listas es entes de ejecutar el programa ya que no son estructuras dinamicas

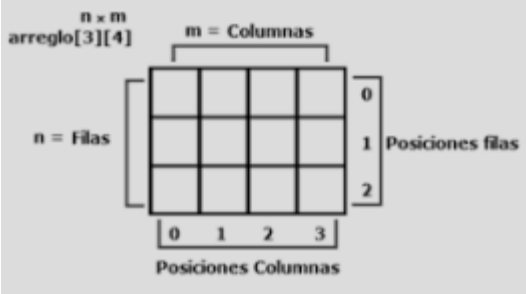

Figura 1. Matriz mediante listas

2. Lista con cabeceras: Mejora el rendimiento y reduce el consumo de memoria solo almacenando solo una coordenada en los nodos. La implementación puede ser mediante una lista simple o doble, en la cual lo elementos son insertado ordenadamente. Cada nodo de la estructura guarda únicamente su valor y los apuntadores son necesarios, no existe ningún desperdicio de memoria y el rendimiento de las operaciones es óptimo.

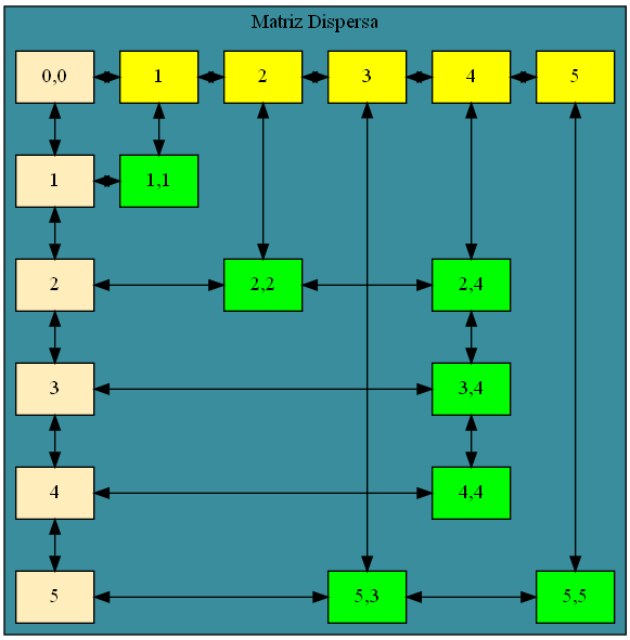

Figura 2. Lita de cabeceras amarillas

3. Lista enlazada: Problemas de rendimiento, en los métodos de inserción (), eliminación() y búsqueda(). Se puede implementar con una lista simple o doble, en el cual los elementos se insertan ordenados. Recalcar que es necesario la utilización de algún método de alineación. Cada nodo de la estructura guarda su posición (x,y), su valor y apuntador a siguiente. Viendo el lado de la memoria es un leve desperdicio por lo que podría ser una opción muy satisfactoria, sin embargo, existen problemas de rendimiento para acceder a los datos

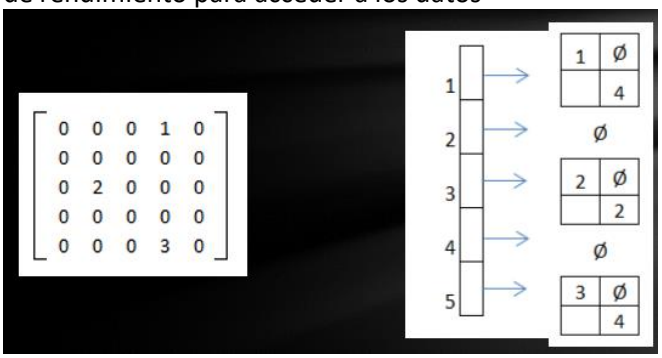

Figura 3. Implementación con lista enlazada

Conclusiones

- Para poder ordenar grandes cantidades de datos en diferentes formatos se aconseja manejar l las estructuras de datos ya que permiten la optimización de memoria ya que las implementaciones nativas requieren de ms memoria de la que comúnmente es usada
- Se recomienda realizar pruebas constantes para tener un punto de referencia para posibles fallas durante la construcción de la solución. Para poder evitar la complejidad de búsqueda de errores
- Conociendo las diferentes formas de implementar una lista se opto por la lista doble ya que permite un recorrido mas personalizado y preciso de los datos además de sus métodos nativos
- Los atributos de filas y columnas, aunque ocupan mas memoria en nuestro programa a comparación de la lista nativa de Python es una implementación que llama menos métodos que la clase original
- El modelo dom es una forma buena de comprender la jerarquía en los grafos ya que

tienen un sistema similar al que los métodos de graficas en forma de arbol

Referencias bibliográficas

Graphviz. (s. f.). The DOT Language. graphviz.org. Recuperado 5 de abril de 2021, de https://www.graphviz.org/doc/info/lang.html pypi.

(2020, 24 diciembre). Graphviz. Recuperado 5 de abril de 2021, de, https://pypi.org/project/graphviz/#description

python.org. (s. f.). xml.etree.ElementTree — The ElementTree XML API — Python 3.9.2 documentation. python. Recuperado 5 de abril de 2021, de https://docs.python.org/3/library/xml.etree.elementtr ee.ht ml

Diagrama de Clases