PROYECTO 1

202001932 – Raúl Josue Castillo Barco

Resumen

Las listas enlazadas son estructuras de datos donde cada nodo está referenciado hacia otro nodo a través de apuntadores. A diferencia de las listas convencionales, no se puede acceder de manera aleatoria.

Estas estructuras de datos son un tipo de dato autorreferenciado porque contiene un apuntador a otro dato del mismo tipo.

El programa analiza celda por celda para verificar si esta se contagia o no. Asi como de verificar si sana. Implementando dichas estructuras.

Al momento de cargar el archivo, toda la información de un paciente es almacenada en una lista simple enlazada.

La matriz principal y listas secundarias se realizaron con dichas estructuras. Para la matriz principal que almacena las células contagiadas se realizó una lista doblemente enlazada para tener acceso tanto a la célula posterior como a la anterior.

Uno de los problemas de las listas enlazadas es el consumo de memoria, pues cada nodo al tener una o más referencias de otros nodos equivale a un espacio en memoria

Palabras clave

- Estructuras de datos
- Memoria principal
- Lista enlazada
- Doblemente enlazada
- Células
- Contagios
- Enfermedades
- Predicción de enfermedad
- Archivos

Introducción

El proyecto está realizado en el lenguaje de programación Python en su versión 3.

Utilizando la programación orientada a objetos se implementaron las distintas clases para cada lista.

La clase principal se encarga de ejecutar todas las funciones del programa tales como: cargar archivo, generar periodos, generar archivo de salida, etc.

El programa da al médico la posibilidad de predecir el estado de algún paciente. Con base a los patrones generados se realizan conclusiones.

El programa permite observar como se comportan los patrones nuevos generados a través de una imagen.

Los resultados se generan en un formato XML.

Desarrollo del tema

Las listas enlazadas son una parte fundamental de las estructuras de datos. Fueron teorizadas desde el año 1956.

La parte fundamental del proyecto es la programación orientada a objetos ya que a partir de este paradigma se generan las listas enlazadas.

Lenguajes como C o C++ permiten un mejor manejo de las punteros o apuntadores.

En este caso implementamos una clase Nodo y como apuntador definimos un atributo de dicho objeto.

El programa contiene 4 listas enlazadas. Tres de ellas simples y 1 doblemente enlazada que es la que permite buscar y analizar los nuevos patrones que se generaran.

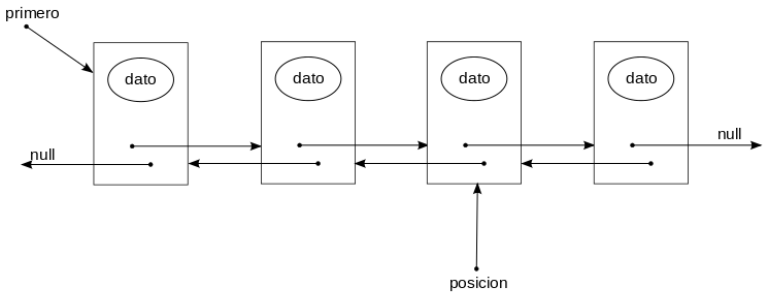
El programa es capaz de realizar periodos de forma automática y detenerse si encuentra un patrón repetido de lo contrario ejecuta la cantidad de periodos solicitados

Las 4 listas son las siguientes:

Datos:
Cuando se carga el archivo de entrada se almacenan en todos los datos de un paciente en un nodo de la lista

Lista doble:
Es la lista principal, en ella se carga el patrón inicial asi como ejecutar según las reglas los nuevos patrones.

Esta implementa un metodo buscar que permite verificar si una celda está o no contagiada por lo cual retorna el nodo si este se encuentra dentro la matriz para su posterior análisis.



De manera grafica se puede observar la estructura de la lista doblemente enlazada. Esto permite analizar la célula posterior y anterior de una célula especifica.

Lista Almacenar:
Cada vez que se genera un nuevo patrón este se almacena en esta lista enlazada, de tal manera

que al agregarse a la lista compara si ya hay un periodo con el mismo patrón. Si es asi, entonces se detiene y no permite agregar más periodos.

```
def append(self,numper,period):
        if self.primero is None:
            self.primero=Nodo(numper,period)
            return

        actual=self.primero
        while actual.siguiente != None and actual.dato!=period:
            actual=actual.siguiente
        if actual.dato==period:
            print("El periodo ya existe")
            self.repetido=True
            actual.siguiente=Nodo(numper,period)
            return numper-actual.numper
        actual.siguiente=Nodo(numper,period)
}
```

Figura 1. Metodo agregar de los distintos patrones de los periodos.
Fuente: Elaboración propia

Lista Salida:
Cuando se determina o no el estado de cada paciente, los resultados se almacenan en esta lista para su posterior uso en la creación del archivo de salida.

Archivo:
Esta clase contiene toda la lógica del programa, desde la carga del archivo hasta la obtención del archivo de salida.

Resolver un problema con la programación orientada a objetos es más sencillo y practico.

Para matrices de gran tamaño, la obtención del último nodo conlleva una cantidad de tiempo considerable ya que debe recorrer toda la lista hasta llegar al último.

La estructura básica del programa es la siguiente:

El nodo raíz o es el archivo main ya que contiene al menú y este hace las llamadas a la clase que contenga la lógica.

El siguiente nivel es el nodo de la clase Archivo que contiene las instancias de las listas utilizadas del último nivel para interactuar con ellas.

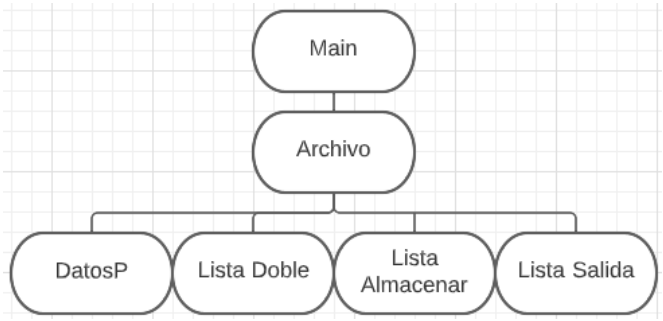


Figura 2. Estructura del programa

Fuente: elaboración propia

Para interactuar con las distintas clases que existían en otro archivo, se utilizó la modularidad, haciendo uso de la instrucción import.

Un módulo no es más que un archivo que contiene funciones, variables y clases.

Una estructura de datos parecida y mucho más rápida es la matriz dispersa, de igual manera se utilizan listas enlazadas, pero es bastante útil para matrices de gran tamaño

Conclusiones

- Las estructuras de datos de listas enlazadas proveen una mayor seguridad ya que no hay forma de acceder a la informacion de un nodo de forma aleatoria.
- Se determino que las listas enlazadas requieren más uso de memoria. Y por lo tanto mayor tiempo de procesamiento.

- Para matrices de gran tamaño una lista enlazada simple o doblemente enlazada generará conflictos.
- Se determino que para una matriz de tamaño mayor que 60 el tiempo de ejecución es considerable.

Anexos

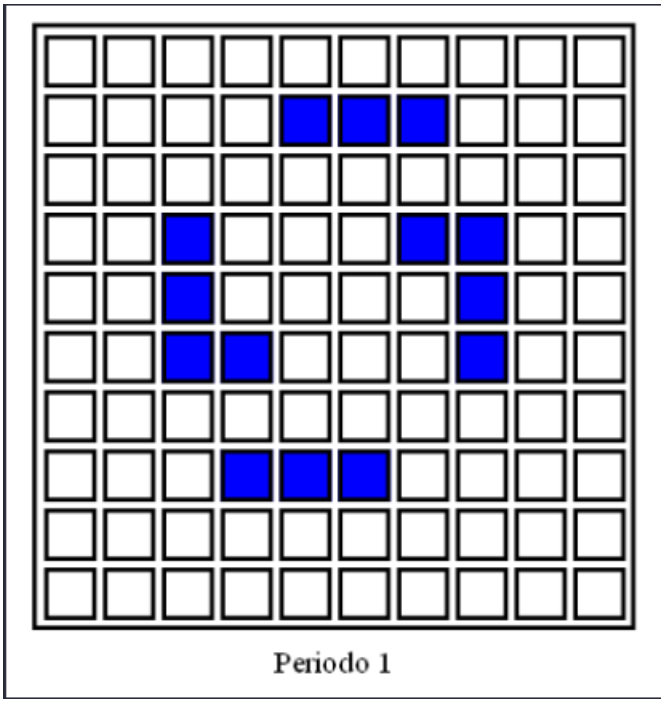


Figura 3. Imagen de salida para un nuevo periodo aplicado.

Referencias bibliográficas

Salcedo, L. (22 de 12 de 2020). *Mi Diario Python* . Obtenido de https://pythondiario.com/2018/07/linked-list-listas-enlazadas.html

Shaik, H. K. (4 de febrero de 2021). *Geekflare*. Obtenido de https://geekflare.com/es/python-linked-lists/