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 2021.

SOLUCION DEL MENOR GASTO DE COMBUSTIBLE MEDIANTE LA UTILIZACION DE LISTAS ENLAZADAS

Carnet 201903553 – Alberto Josué Hernández Armas

Resumen

Para solucionar el problema planteado se utilizó un método de estracción de información de un archivo de extensión xml, esta información se trabajó por medio de listas enlazadas para un manejo más eficiente. Luego de obtenerla en las listas se muestra también la relación entre los nodos mediante la herramienta graphviz, luego se procede a ejecutar el algoritmo para encontrar el camino más corto mediante el algoritmo Dijsktra. Este algoritmo hace una etiqeutación con base en el nodo incial, determinando la distancia entre cada nodo, siendo este la suma de la cantidad necesaria de los nodos vecinos, luego hace una regresión mediante la ruta de nodos etiquetados, para definir las rutas más cortas de todos los nodos hacia el nodo inicial. de estas rutas se selecciona la cual donde los nodos en los extremos son los definidos como inicial y final en el archivo de entrada

Luego se grafica esta relación, mostrando una matriz, ya no con los valores de entrada, sino solamente como posiciones marcadas de donde debe pasar el robot.

Palabras clave

- Lista enlazada
- Estructura de datos
- Algoritmo Djkstra
- XML
- Nodos

Abstract

To solve the given problema, there was an informatioin extraction methon implemented to a given xml file that contained all the data. This information was worked through linked lists for a more efficient handling. After obtaining it in the lists, the relationship between the nodes is also shown using the graphviz tool, then we proceed to execute the algorithm to find the shortest path using the Dijsktra algorithm. This algorithm makes a labeling based on the initial node, determining the distance between each node, then it makes a regression through the route of labeled nodes, to define the shortest routes of all nodes towards the initial node. From these routes, the one where the nodes at the ends (the first and the last) are those defined as initial and final in the XML input file is selected.

Then this relationship is graphed, showing a matrix, no longer with the input values, but only as marked positions from which the robot must pass.

Keywords

- *Linked List*
- *Data structure*
- *Djkstra Algorithm*

- *XML*
- *Node*

.

Introducción

Para el manejo de información o una serie de datos, no pueden trabajarse como un objeto individual, se deben accesar de manera ordenada y teniendo una distinción clara entre toda la información proporcionada. Para esto necesitamos proporcionarle una estructura. Existe una cantida enorme de tipo de estructuras: Listas, Matrices, Listas enlazadas, Listas, Colas, etc. En este proyecto se nos presenta un robot que desea atravesar un terreno, gastando la menor cantidad de combustible posible, que representa su terreno mediante un flujo de información en un archivo de tipo XML. La implementación de la solución se hizo con base en la estructura de Listas Enlazadas, para poder construir una relación entre todos los nodos y así implementar el algoritmo de encontrar el camino más corto(se seleccionó el algoritmo Dijkstra), haciendo más rápido el etiquetado de los nodos. Toda esta implementación continua de algoritmos y estructuras permite el correcto funcionamiento del programa teoría y metodología anteriormente mencionada.

Debe contener un máximo de 150 palabras.

Desarrollo del tema

Estructura de Datos

Un dato simple, no contiene estructuras dentro de este, tiene una representación directa, pero al estar en conjunto, genrealmente de una forma en la que podría decirse que están enlazadas, toman una estructura, o almenos es necesario dárselas. Esta estructura se caracteriza por su colección de datos simples y la lógica operacional que se define ente estos. Estos pueden ser objetos con representación con una estructura definida por el programador. En este proyecto se desarrollan de la manera de Listas enlazadas, este tipo de estructura es una cadena de objetos apuntando en una misma dirección hasta llegar a la nada. Se declara un objeto llamado nodo, el cual contendrá un objeto específico de la información proporcionada, es decir, el nodo es el objeto con la representación inmediata. Se le agrega un atributo, el cual es un nodo ya existente, generalmente en este se agrega el que va siguiente, se le denomina apuntador. Esta secuencia de nodos es lo que transforma la información individual en una Lista Enlazada. La ventaja de este tipo de estructura es que el tipo de dato no tiene que ser inmutable, Los nodos al ser tipo objeto pueden pertenecer a cualquier tipo de dato.

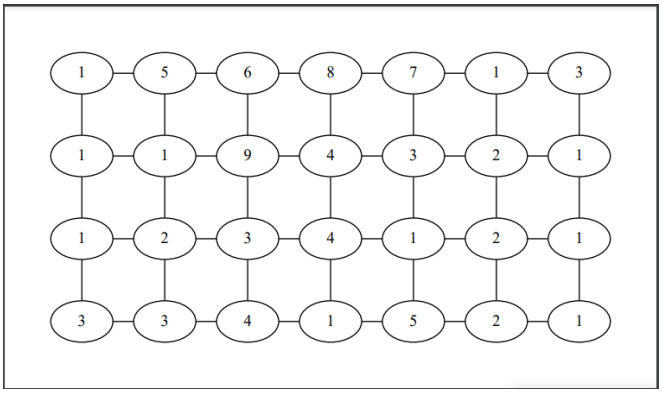

Figura 1. Ejemplo del enlazamiento de nodos de una Lista Enlazada

Fuente: elaboración propia

Ya que no existe una forma directa para recorrer este tipo de estructura, es necesario general algoritmos

que nos lo permitan, la implementación y flexibilidad de estos depende del entorno que se desee desarrollar. Al ser una lista simplemente enlazada, se puede aplicar una iteración que detecte la existencia de nodos hasta llegar a null. Mientras esta iteración esta en proceso se aprovecha a obtener los datos existentes en esta estructura.

A continuación un ejemplo:

```
while ahora != None and -1 == str(ahora).find("a") and -1 == str
  minuevalista.agrega(ahora)
  mpxaux.agrega(ahorax)
  mpyaux.agrega(ahoray)
  ahora = ahora.Next
  ahorax = ahorax.Next
  ahoray = ahoray.Next
```

Figura 2: ejemplo de algoritmo de iteración de Lista enlazada
Fuente: Elaboración propia

Esto es si la estructura ya esta definida, pero el caso es que es una estructura flexible, lo cual nos permite agregar datos. En este proyecto, la cadena de información proporcionada se va agregando uno por una a nuestra cadena de datos de Lista Enlazada. La forma en la que esto se realiza es mediante la instancia de un nuevo nodo, y este será agregado en la cabeza de la lista. Será quien apunte al último nodo agregado antes que el. Por o que el orden es el siguiente: si la cadena esta vacia, la propiedad .Next del nuevo nodo a agregar será la propiedad Null. Y si la cadena tiene datos. La propiedad .Next del nodo será el último nodo agregado a la lista, el que se encuentra en la cabeza. Este método iterativo nos permite crear una cadena en una sola dimensión y dirección, ya que solo podrá recorrerse hacia abajo, desde el último que se agregó hasta el nodo con valor nulo.

```
class Node:
    def __init__(self,Value):
        self.Value = Value
        self.Next = None

    def __str__(self):
        return str(self.Value)
```

Figura 3: ejemplo de estructura de un nodo en lenguaje pytohn
Fuente: Elaboración propia

```
def agrega(self,Value):
    mnodo = Node(Value)
    if self.Size == 0 :
        self.First = mnodo
    else:
        ahora = self.First
        while ahora.Next != None:
            ahora = ahora.Next
        ahora.Next = mnodo

    self.Size += 1
    return mnodo
```

Figura 4: ejemplo de la función que enlaza los nodos en una dirección
Fuente: Elaboración propia

Ahora con los nodos obtenidos se requiere de un algoritmo de implementación que permita saber el camino más corto. El algoritmo Dijkstra, covnerte la matriz en nodos eiquetados, donde el valor de la distencia inmediata es la suma de los nodos valor en lso extremos el trabjo a etiquetar. En sí, dos nodos forman una etiqueta, pero estas etiquetas conservan la estructura de la Lista Enlazada. La etiqueta se refiere el nodo proveniente y la distanca(en este caso combustible necesario para recorrer este camino). Esta etiquetación continúa hasta el final del

enlazamiento, definiendo una nueva estructura, totalmente dependiente de la primera.

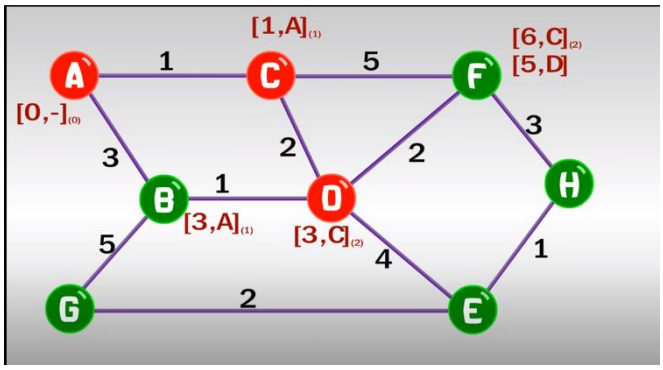

Figura 4: ejemplo de la función del algoritmo Dijkstra
Fuente: Studios Cat(canal de youtube)

CONCLUSIONES

- Las estructuras de datos permiten un manejo eficiente de la información
- El tipo de Estructura de datos implementada dependen del entorno y lo que se desee hacer con la información
- Las Listas enlazadas simples son una forma compleja de realizar una estructura de información pero son más dinámicas y flexibles en comparación con listas u otras estructuras
- El algoritmo Dijkstra a pesar de estar restringido por un tipo de entrada específico y un ecosistema específico, es bastante eficiente, ya que genera todas las posibles soluciones respecto a un nodo sin agregar ambigüedades como caminos repetidos o caminos más largos erróneamente agregados

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems.* Addison-Wesley Publishing Company, Inc.

Studios Cat. (2020). Dijkstra. 2021, de Studios Cat Sitio web: https://www.youtube.com/watch?v=LLx0QVMZVkk

anonimo. (2020). Estructura de Datos. 2021, de Anonimo Sitio web: http://informatica.uv.es/docencia/fguia/TI/Libro/PDFs/CAPI5.pdf

Victor Sanz. (2020). Implementacion Algoritmo Dijkstra. 2021, de Victor Sanz Sitio web: https://www.youtube.com/watch?v=w84P5Coifek