
PROYECTO 1 - MINWAY

202003975 – Diego André Mazariegos Barrientos

Resumen

El software denominado “Proyecto 1 MinWay” el cual es una aplicación enfocada en apoyar en la tarea del cálculo del recorrido que consume la mínima cantidad de combustible en un determinado terreno esto con el fin de apoyar al robot r2e2 diseñado por la Agencia Guatemalteca de Investigación Espacial (AGIE) en sus diversos recorridos.

El software fue desarrollado mediante la implementación de Tipos de Datos Abstractos (TDA's) para el almacenamiento de los datos y su manipulación, el uso del algoritmo de Dijkstra junto con teoría de grafos para determinar el camino óptimo, manipulación de archivos con extensión XML para el ingreso de datos, creación de reportes mediante la tecnología Graphviz y el uso de estructuras recursivas para la optimización del rendimiento del producto.

El proyecto solicitado cumple con todos los requerimientos propuestos por el cliente, se cumplieron y se probaron profundamente todas las opciones solicitadas.

Palabras clave

- TDA, Dijkstra, teoría de grafos, Graphviz, memoria dinámica

Abstract

The software called "Proyecto 1 MinWay" is an application focused on supporting the task of calculating the route that consumes the minimum amount of fuel on a given terrain, to support the r2e2 robot designed by the "Agencia Guatemalteca de Investigación Espacial" (AGIE) in its various routes.

The software was developed by implementing Abstract Data Types (ADT's) for data storage and manipulation, the use of the Dijkstra algorithm together with graph theory to determine the optimal path, manipulation of files with an XML extension for the data entry, report creation using Graphviz technology and the use of recursive structures to optimize product performance.

The requested project meets all the requirements proposed by the client, all the requested options were fully met and tested.

Keywords

- TDA, Dijkstra, graph theory, Graphviz, dynamic memory.

Introducción

El presente artículo tiene como finalidad dar un análisis profundo acerca del desarrollo del software “Proyecto 1 MinWay”.

El software diseñado se basa principalmente en el paradigma de la programación orientada a objetos, esto con el fin de darle una forma organizada al programa, para conservar la simpleza del mismo y si en algún punto se necesitase la reutilización del mismo.

Por medio de la programación orientada a objetos se hizo posible la implementación de los Tipos de Datos Abstractos (TDA's), los cuales fueron de utilidad para el guardado y manipulación de los datos entrantes y salientes del programa por medio de archivos de extensión XML.

Para el procesamiento de los datos se hizo uso de estructuras recursivas y mediante la teoría de grafos se aplicó el algoritmo de Dijkstra para la búsqueda del camino que consumiese menor cantidad de combustible, para la impresión se utilizó la tecnología Graphviz.

Desarrollo del tema

El contexto del proyecto se basa en la necesidad del robot r2e2 de suministrar el combustible almacenado y la capacidad que este tiene para comunicarse con el satélite Quetzal01 para obtener una imagen del terreno que explorara, por lo que los requerimientos del cliente fueron desarrollar un software que fuese capaz de calcular la ruta que utilice la mínima cantidad de combustible que necesitase para llegar desde su punto de inicio hasta su punto destino con un límite de combustible almacenado de 9999 unidades. El movimiento del robot r2e2 es ortogonal lo que quiere decir que únicamente se puede mover hacia arriba, abajo, izquierda y derecha, debido a esto el mapa del terreno se interpreta como una matriz de ‘x’ filas y ‘y’ columnas, como se muestra en la Figura A1.

El programa cuenta con un menú en el cual están disponibles 6 opciones como se muestra en la figura A2. Para la carga de archivos se realiza por medio de archivos con extensión XML que cuentan con una estructura previamente definida como se muestra en la Figura A3. La lectura e interpretación de dicha estructura se realiza mediante la interfaz de programación de manipulación XML, ElementTree. La biblioteca ElementTree incluye herramientas para analizar XML usando API's basadas en eventos y documentos. Esta misma estructura y herramientas se utilizan para la generación de los archivos de salida. Para el almacenamiento de los datos se utilizan los TDA's y para la generación de los reportes la tecnología Graphviz. Procedimientos los cuales se detallarán con más profundidad a continuación.

a. Almacenamiento de los datos de entrada.

Los datos de entrada que serían una determinada cantidad de terrenos se almacenan en una lista enlazada simple que guarda la información individualmente de cada terreno en forma de objeto como en la Figura 1.

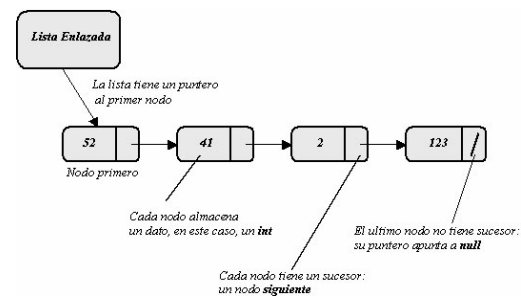


Figura 1. Lista enlazada simple.

Fuente: Estructura de datos, UBO, 2021.

b. Almacenamiento de las posiciones de cada terreno.

Cada objeto/terreno en la lista de terrenos cuenta con una lista ortogonal que sería la matriz de posiciones en cada terreno formada por dos listas enlazadas dobles que funcionan como cabeceras en una matriz ortogonal dinámica, se dice que esta matriz ortogonal es

dinámica ya que no se define el tamaño de esta directamente en el programa, sino que su tamaño se acopla a la cantidad de posiciones ingresadas en cada terreno en el tiempo de ejecución ver la Figura 2.

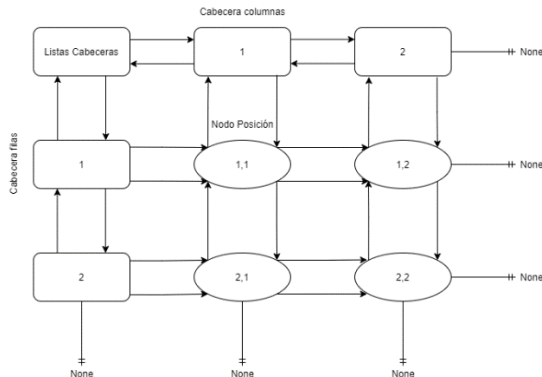


Figura 2. Matriz ortogonal formada por dos listas doblemente enlazadas por cabeceras.

Fuente: elaboración propia.

c. Algoritmo de Dijkstra.

Algoritmo fundamental en el proyecto que consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices, etiquetando cada nodo que va recorriendo de modo que así compara mediante las etiquetas de cada nodo cual es el camino ideal. Cuando se obtiene el camino más corto desde el vértice origen hasta el resto de los vértices que componen el grafo, el algoritmo se detiene y mediante las etiquetas anteriormente descritas va recorriendo el camino en reversa para obtener el camino más corto. Cabe recalcar que este algoritmo funciona únicamente hacia adelante por lo que no se puede retroceder en ningún camino, además no funciona con grafos con aristas negativas ya que, al basarse únicamente en la distancia menor de cada nodo, al encontrarse con una arista negativa este ignoraría cualquier ruta alterna que pudiese darse el caso fuese un mejor

camino. El algoritmo funciona de la siguiente manera como se muestra en la Figura 3.

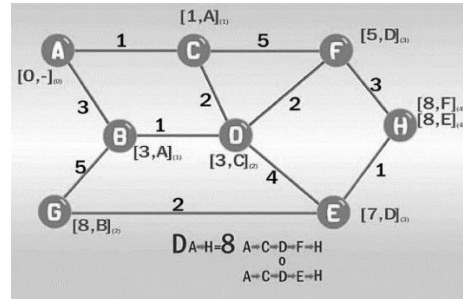


Figura 3. Matriz ortogonal formada por dos listas doblemente enlazadas por cabeceras.

Fuente: Algoritmo de Dijkstra, gvital, VIMP, 2021.

d. Archivos de salida.

El programa cuenta con una estructura ya definida para los archivos de salida, así como para los de entrada su extensión es de tipo XML su estructura sería la siguiente ver Figura 4, en donde los cambios respectivos a los archivos de entrada es que en el apartado de combustible se hace referencia al valor del consumo por el terreno procesado y las posiciones mostradas son el camino mínimo desde el punto de origen al de destino.

```
<?xml version="1.0"?>
- <terreno5>
  - <posicioninicio>
    <x>7</x>
    <y>7</y>
  </posicioninicio>
  - <posicionfin>
    <x>13</x>
    <y>1</y>
  </posicionfin>
  <combustible>42</combustible>
  <posicion y="7" x="7">5</posicion>
  <posicion y="7" x="8">1</posicion>
  <posicion y="7" x="9">5</posicion>
  <posicion y="7" x="10">3</posicion>
  <posicion y="6" x="10">2</posicion>
  <posicion y="5" x="10">2</posicion>
  <posicion y="4" x="10">2</posicion>
  <posicion y="4" x="11">7</posicion>
  <posicion y="3" x="11">5</posicion>
  <posicion y="2" x="11">1</posicion>
  <posicion y="1" x="11">1</posicion>
  <posicion y="1" x="12">1</posicion>
  <posicion y="1" x="13">7</posicion>
- </terreno5>
```

Figura 4. Estructura archivo de salida.

Fuente: elaboración propia.

e. Sistema de reportes.

Para la generación de reportes y así poder visualizar de forma más agradable los terrenos se utiliza la tecnología Graphviz y se trabaja con archivos de salida de tipo DOT los cuales son legibles por medio de Graphviz que ofrece una interfaz para interpretar estos documentos en formato PNG dando como resultado los siguientes archivos de salida ver Figura 5.

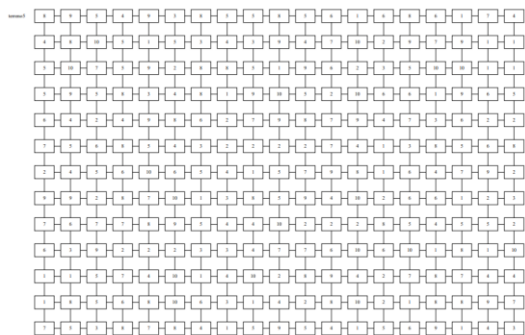


Figura 5. Reporte de terreno generado mediante la herramienta Graphviz.

Fuente: elaboración propia.

Para una mejor comprensión del programa y de los temas desarrollados ver Figura A4, donde se expone de una forma más clara como están relacionadas todas las funcionalidades del proyecto.

Conclusiones

El proyecto realizado es óptimo y cumple con las especificaciones requeridas por el cliente por lo que la aplicación fue completada con éxito.

Es posible y viable implementar el lenguaje de Python para dar una solución al planteamiento presentado.

Graphviz es una herramienta útil que permite la generación de reportes para gráficas relacionadas con grafos.

El Algoritmo de Dijkstra es funcional y puede implementarse junto con TDA's y aplicar esto en la memoria dinámica como se demostró en este trabajo para la determinación de un camino óptimo en una diversa cantidad de rutas.

Referencias bibliográficas

C. R. Severance, (2016). Python para todos. Recuperado en el año 2021. Disponible en https://uedi.ingenieria.usac.edu.gt/campus/pluginfile.php/145984/mod_resource/content/1/pythonlearn.pdf

M. Yassin, (2021). Python for beginners - Learn all the basics of python. Recuperado en el año 2021. Disponible en <https://www.udemy.com/course/python-for-beginners-learn-all-the-basics-of-python/>

R. P. Grimaldi, (1994). Matemática discreta y combinatoria: Una introducción con aplicaciones. (3ª ed.), Massachusetts, Addison-Wesley Publishing Company, Inc.

S. Bank, (2013). Graphviz. Recuperado en el año 2021. Disponible en <https://graphviz.readthedocs.io/en/stable/index.html>

Apéndices

Fila / Columna	1	2	3	4	5
1	1	1	5	3	2
2	4	1	4	2	6
3	3	1	1	3	3
4	5	2	3	1	2
5	2	1	1	1	1

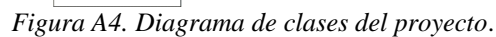
Figura A1. Terreno de exploración recibido desde el satélite Quetzal01.



```
<terreno>
  <terreno nombre="terreno1">
    <posicioninicio>
      <x>1</x>
      <y>1</y>
    </posicioninicio>
    <posicionfin>
      <x>2</x>
      <y>2</y>
    </posicionfin>
    <posicion x="1" y="1">1</posicion>
    <posicion x="1" y="2">1</posicion>
    <posicion x="2" y="1">34</posicion>
    <posicion x="2" y="2">3</posicion>
  </terreno>

  <terreno nombre="terreno2">
    <posicioninicio>
      <x>1</x>
      <y>1</y>
    </posicioninicio>
    <posicionfin>
      <x>1</x>
      <y>2</y>
    </posicionfin>
    <posicion x="1" y="1">1</posicion>
    <posicion x="1" y="2">1</posicion>
    <posicion x="2" y="1">5</posicion>
    <posicion x="2" y="2">1</posicion>
  </terreno>
</terrenos>
```

Fuente: elaboración propia.



Fuente: elaboración propia.