UNIVERSIDAD PRIVADA BOLIVIANA

Proyecto Final Algorítmica I

“CoDyMe_HospitalSearcher”

ESTUDIANTES: **Camila Alejandra Grandy Camacho**
Joseph Anthony Meneses Salguero
Marcos Andres Simon Agreda

CURSO: **Algorítmica I**

PROF. DE LA MATERIA **Ing. Paul Wilker Landaeta Flores**

LA PAZ 14/05 – BOLIVIA
2021

RESUMEN

El presente informe, muestra en su totalidad todos los pasos realizados para poder desarrollar un simulador de búsqueda de hospitales, llamado: “CoDyMe_HospitalSearcher”, cuyo funcionamiento está enteramente basado en el famoso algoritmo creado por Edsger W. Dijkstra para poder encontrar los caminos mínimos de un grafo con pesos positivos en las aristas; conocido como: El algoritmo de Dijkstra. El objetivo del programa es, a partir de un grafo, que representa un continente; viajar a través de los vértices del grafo, representando las capitales de los países que contienen un hospital; para obtener las distancias mínimas para poder llegar al hospital deseado. Este proyecto se formuló en su totalidad en el lenguaje de programación C++.

ABSTRACT

This report, shows in its totality, all the steps performed to develop a hospital searcher simulator, named: “CodyMe_HospitalSearcher”, whose operation is entirely based on the famous algorithm created by Edsger W. Dijkstra, which finds the shortest paths in a non-negative weighted graph; known as: Dijkstra’s Algorithm. The purpose of the program is, beginning from a graph, which represents a continent, traverse through all the nodes of the graph, which represent the capitals of the countries that contain a hospital, to be able to obtain all the minimum distances between the starting point and the selected hospital. This project was written completely in the programming language C++.

INTRODUCCIÓN

Hoy en día, especialmente debido al desarrollo de la pandemia mundial que está atravesando la humanidad, las conexiones entre hospitales se han visto muy debilitadas, y todo por culpa de la casi nula organización que los gobiernos están aplicando al sistema sanitario mundial; además, muchos hospitales están tan insuficientemente equipados, que muchas veces es necesario transferir a los pacientes a otros hospitales; y en el peor caso de que todo el país esté completamente desestructurado, mover al paciente a otro país se vuelve una necesidad; pero, debido a que muchas veces la condición de los pacientes es crítica, existen solo algunos caminos por los cuales los pacientes pueden ser transportados, los cuales, para poder preservar la integridad de los pacientes, deben ser los más cortos.

Por esto mismo, fue que el grupo se decidió por crear la herramienta: "CoDyMe_HospitalSearcher", una herramienta escrita en su totalidad en C++, la cual basa su funcionamiento en el conocido algoritmo de Dijkstra, el cuál puede encontrar los caminos más cortos en un grafo. La herramienta consiste en introducir un punto de partida, y uno de llegada, simbolizando el hospital de partida y el hospital de llegada; y posteriormente, la herramienta conseguirá el camino más corto entre ambos hospitales, facilitando la transportación del futuro paciente.

1. PROCESO DE INSTALACIÓN. –

- Descargar e instalar WinRar
- Descargar la última versión disponible de Visual Studio Code
- Seguir las instrucciones de instalación que proporciona el instalador de Visual Studio Code.
- Descargar el compilador de C++11, llamado g++, mediante la descarga del instalador de mingw.
- Descomprimir el instalador, usando WinRar, en la ruta deseada.
- Seguir las instrucciones de instalación del paquete que contiene el compilador. Recuerde la ruta de instalación.
- Dirigirse a **Panel de control\Sistema y seguridad\Sistema** y apretar el botón **Configuración avanzada del sistema**.
- En la sección de **Opciones avanzadas**, hacer click en **Variables de entorno**.
- En la parte de **Variables del sistema**, seleccionar la opción **Path** y hacer click en **Editar**.
- Presionar el botón **Nuevo** e indique la ruta de instalación del compilador mingw, y haga click en **Aceptar**.
- Descargue el contenido del repositorio de GitHub.
- Descomprima el archivo .zip utilizando WinRar.
- Coloque los archivos en una carpeta deseada.
- Abra Visual Studio Code
- Dirigirse a la pestaña de Extensiones e instale la extensión C/C++
- En la parte superior, hacer click en **Archivo** y **Abrir Folder**
- Seleccione la carpeta que contenga los archivos

- Diríjase al archivo **CoDyMe_HospitalSearcher.cpp** y compile el archivo en la sección de **Compilar y debuguear**.

2. DEFINICIÓN DEL PROBLEMA –

Debido al contexto mundial que está sufriendo la humanidad, por culpa del ya conocido virus Covid-19; el sistema de salud en la mayoría de países ha sufrido grandes percances en el ámbito, tanto humanitario, como administrativo. Concentrándose más en este último ámbito, debido al crecimiento desproporcionado del virus antes mencionado, muchos de los hospitales principales de los países más grandes de todo el mundo, se han visto completamente abrumados por la gran cantidad de personas acudiendo a estos centros, buscando ayuda.

Esta situación se ha visto todavía más agravada, si se toma en cuenta, a los países más pobres de los seis continentes, los cuáles, lejos de tener un sistema de salud lo más mínimamente estable; se encuentran en una verdadera crisis sanitaria, la cual atenta casi diariamente con las vidas de los ciudadanos, los cuales no pueden recibir una atención sanitaria con un mínimo de calidad.

Por esto mismo, la solución más lógica, para estos países poco desarrollados, es mover a los pacientes a los hospitales que en mejor condición estén, (obviamente, esto implica una corta distancia, es decir, hospitales en un mismo continente) y así poder brindarles un tratamiento médico de calidad.

Lamentablemente, el verdadero problema, radica en el hecho de que, muchas veces los pacientes deben ser trasladados en el menor tiempo posible, debido a que muchas veces los pacientes suelen estar en un estado crítico de salud; por lo que ahorrar, aunque sea un solo minuto en transporte, puede significar la diferencia entre la vida y la muerte de un paciente.

Considerando la gravedad del problema, fue por lo que diseñó y creó la herramienta "CoDyMe_HospitalSearcher", la cual está enteramente basada en el algoritmo de Dijkstra.

3. EXPLICACIÓN DEL ALGORITMO. –

El algoritmo de Dijkstra, también llamado algoritmo de caminos mínimos, es un algoritmo para poder determinar el camino más corto, dado un vértice origen, hacia el resto de los vértices en un grafo que tiene pesos en cada arista. Cabe destacar que éstos pesos deben ser positivos, ya que el funcionamiento del algoritmo, se rompe completamente, si es que se usan pesos negativos.

Ahora, se describirán los pasos que sigue el algoritmo de Dijkstra:

- Declarar tres estructuras: Una cola de prioridad, un arreglo que guarde las distancias entre vértices y un arreglo que guarde los vértices, si ya han sido visitados.
- En el arreglo de visitados, marcar todas las casillas correspondientes a los vértices con infinito (un valor muy grande).
- En el arreglo de visitados, marcar la distancia del vértice inicial con 0.
- Visitar a los vecinos del vértice actual.
- Calcular la distancia del vértice vecino.
- Verificar si la distancia es mínima, si es así, guardar la nueva distancia.
- Repetir los pasos anteriores, hasta haber visitado todos los vértices.

Siguiendo los pasos del algoritmo de Dijkstra, el pseudocódigo estaría descrito de la siguiente manera:

```
1  function Dijkstra(Graph, source):
2      dist[source] ← 0
3
4      create vertex priority queue Q
5
6      for each vertex v in Graph:
7          if v ≠ source
8              dist[v] ← INFINITY
9              prev[v] ← UNDEFINED
10
11         Q.add_with_priority(v, dist[v])
12
13
14     while Q is not empty:
15         u ← Q.extract_min()
16         for each neighbor v of u:
17             alt ← dist[u] + length(u, v)
18             if alt < dist[v]
19                 dist[v] ← alt
20                 prev[v] ← u
21                 Q.decrease_priority(v, alt)
22
23     return dist, prev
```

Esto daría como resultado, una complejidad algorítmica de:

O(n) = V+E*LOGV

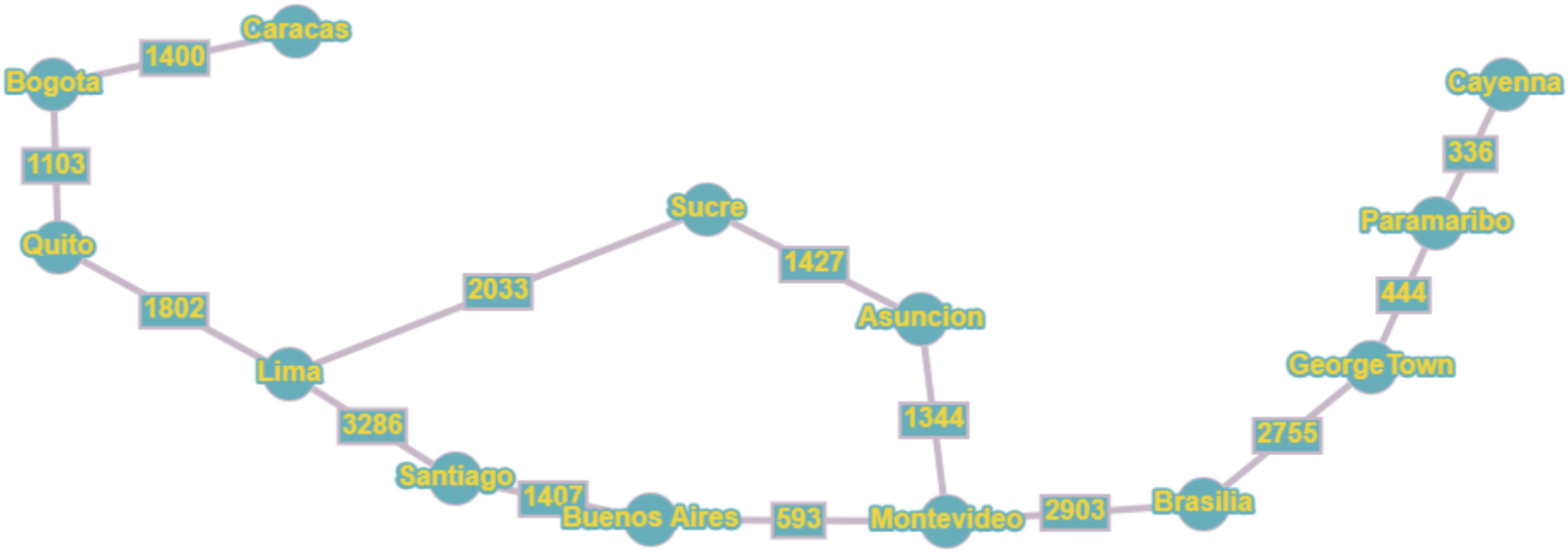
Sabiendo la importancia y la gran capacidad de este algoritmo, se lo aplicó como la base del presente proyecto. El proyecto tiene como tarea inicial, el buscar las distancias mínimas entre las capitales de dos distintos continentes (tres, si se separa Europa del Oeste de Europa del Este), ya que en las capitales de los países es dónde se encuentran los mejores hospitales; pero a la vez, estos hospitales suelen estar bastante llenos, por lo que seguramente se tendrá que cambiar de capital más de una vez.

El algoritmo de Dijkstra, se aplica, en la búsqueda de caminos entre las capitales, en este caso, entre los caminos más cortos entre las capitales. Usando la explicación antes dada, del algoritmo de Dijkstra, se proporcionará a la aplicación un vértice inicial, el cual representará el punto de inicio para el grafo, el algoritmo realizará la búsqueda de las distancias más cortas en todo el grafo, para luego imprimir la distancia deseada (esto se especificará con un punto de

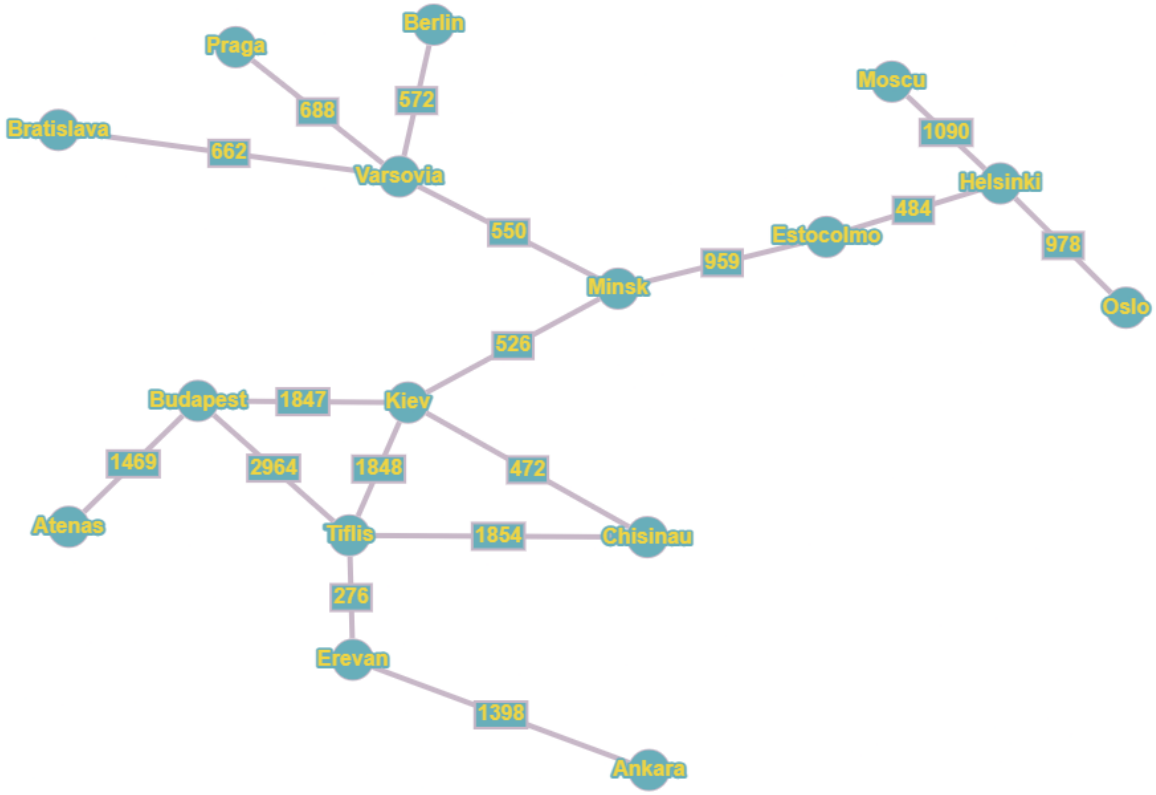
llegada). Finalmente, si el usuario desea realizar otra consulta, el programa eliminará todos los contenidos del grafo, el arreglo de distancias y el arreglo de visitados; para que así, el algoritmo esté libre para poder hacer otra consulta.

Para concluir la explicación del algoritmo, se proporcionarán los grafos que se utilizaron en el proyecto:

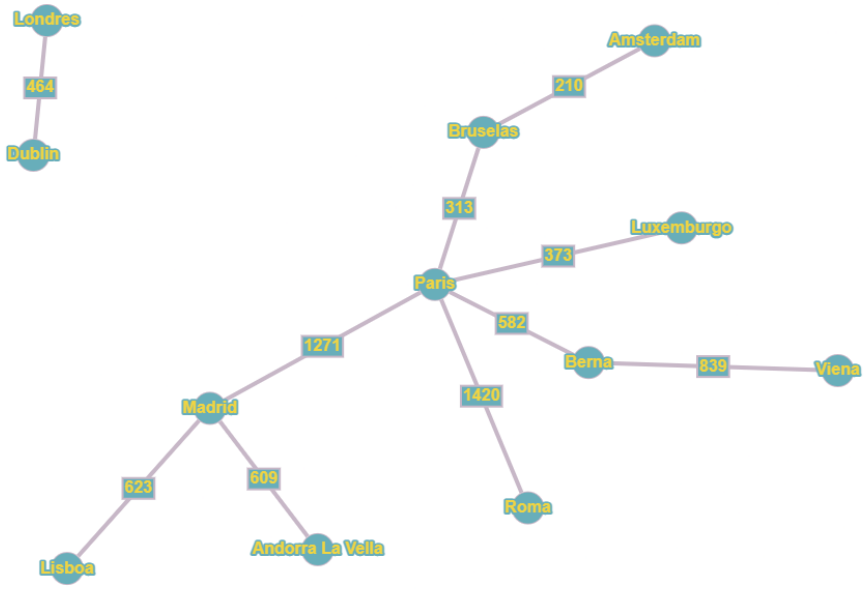
1. América Del Sur. –



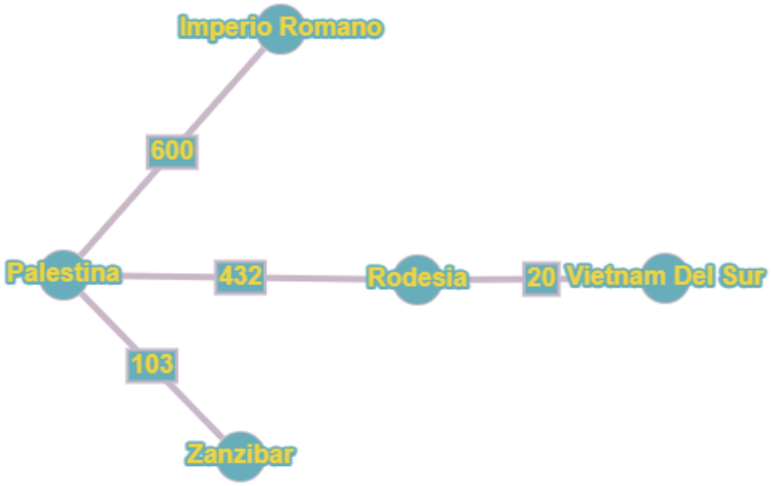
2. Europa del Este. –



3. Europa del Oeste. –



4. Ejemplo 1 (Lemuria). –



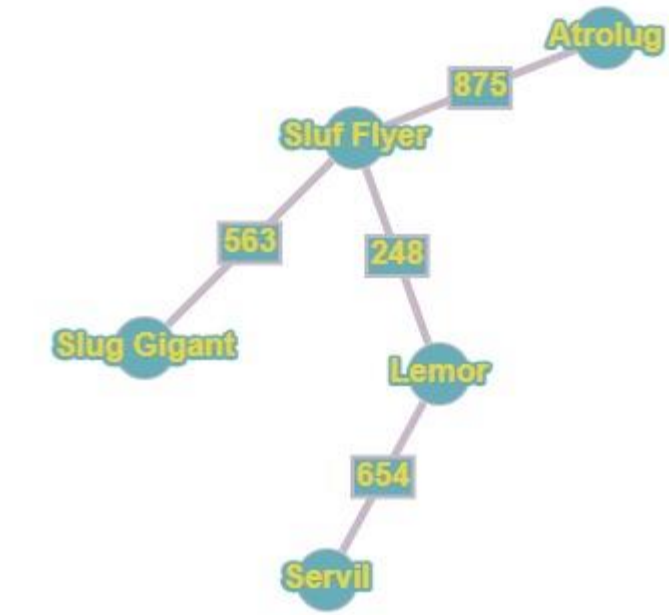
5. Ejemplo 2 (Atlántida)

6. Ejemplo 3 (Wakanda). –

7. Ejemplo 4 (Zootopia). –

8. Ejemplo 5 (GOW). –

9. Ejemplo 6 (Meslu). –



4. CONCLUCIONES-

El buscador de hospitales, CoDyMe_HospitalSearcher, ayuda de gran magnitud a todos los clientes, ya que ayuda a encontrar el camino desde el hospital que se encuentre hasta el que desee de manera eficaz.

Esta ventaja, es posible gracias al algoritmo de Dijkstra y la facilidad que tiene para encontrar los caminos más cortos entre los hospitales; además esto facilita el trabajo de tener que hacer estos cálculos manualmente y tardar más tiempo teniendo la posibilidad de caer en errores debido a la cantidad de distancias que se maneja.

Hoy en día, este tipo de problemas lo podemos ver con mayor frecuencia, ya que, con la creciente pandemia, varias personas se han visto en la necesidad de buscar otros hospitales que, además de tener la especialidad buscada, tengan la posibilidad de informar al cliente/paciente con los detalles más precisos sobre su tratamiento, lo cual, es algo bastante inusual hoy en día, debido a que los hospitales suelen estar bastante saturados, lo cual impide que el personal del hospital pueda darle una atención de calidad a TODAS las personas.