Introducción

- Presentar a los integrantes

- Presentar de que trata el programa

Diapositiva #2

Introduccion: Este trabajo fue desarrollado en 3 lenguajes distintos, tenemos a c++ que es un lenguaje que ya conocemos, por otro lado MS-DOS que es un lenguaje nativo de Windows que ultilizamos para solucionar algunos problemas que se nos fueron presentando, y por ultimo tenemos a DOT que es el lenguaje que utiliza el Graphviz (software que grafica diagramas), en este caso nosotros usamos el grafo

Diapositiva #3

En primer lugar vamos a explicar los diferentes codigos que hicimos en c++.

Comenzamos explicando el primer cpp donde este se encarga de la creación del grafo.

Este código lo que hace es crear las matrices de adyacencia, una que almacena las distancias que tienen las aristas y otra almacena los nombres de las calles.

Para esto le mandamos como parámetros …

Diapositiva #4

En esta función guardamos las dos matrices creadas anteriormente en unos .txt.

Diapositiva #5

Por otro lado tenemos otro cpp que es el que se encarga de realizar los recorridos

Aca se ve el código que lee los txt creados anteriormente y asigna los datos a las matrices de adyacencia.

Diapositiva #6

Este seria el mapa del grafo de las dos matrices de adyacencia

Diapositiva #7

El usuario ingresaría las direcciones a las que tiene que ir. Para ello utilizamos una lista enlazada.

Los parámetros que se le mandan a la función son….

Diapositiva #8

Ahora arrancamos con el código que va calculando los recorridos minimos.

Lo que hacemos en un bucle for de h cantidad de direcciones que tenemos….

Diapositiva #9

Después en el bucle “do” se guardaría en una lista enlazada todo el recorrido a partir del vector de padres de la función Dijkstra.

Diapositiva #10

Algoritmo de Dijkstra donde calcula la distancia mínima desde el origen que le mandamos como parámetro con todos los nodos del grafo

Diapositiva #11

Agrandar imagen

En esta función se crea el archivo de texto con la información que necesita en graphviz para que represente el grafo. En el while hace todo esto.

Por ultimo en la parte de abajo se puede ver que se llaman diferentes archivos que va a explicar mati ahora.

Diapositiva #12

Bueno ahora quiero comentarles un poco como funciona el Graficador que elegimos, y es que este software es tan sencillo como que lee un .txt con aristas escritas entre llaves, a las que se les puede poner atributos, como por ejemplo a la primer arista le pusimos un color azul y una etiqueta “Arista 1”.

Existen más atributos, como ordenar el grafo de izquierda a derecha o cosas así visuales, pero no lo necesitamos para este trabajo.

Diapositiva #13

Al crear el documento .txt y quererlo graficar con el software Graphviz nos ddimos cuenta que teníamos un pequeño error de sintaxis. nuestro archivo debia quedar como el de la derecha (digamos con comillas en los nombres), Esto no lo podíamos solucionar ya que c++ no nos permite poner comillas simples con comillas dobles, por lo que tuvimos que recurrir a un script en lenguaje MS-DOS para solucionar esto ¿Cómo lo hicimos? Con el arreglador 1 y 2 que fede mostró antes

Diapositiva #14

Para esto hicimos ciertos comandos, primero desactivamos lla salida para que el usuario no vea el script, segundo nos aseguramos de crear un archivo vacio (eliminando si existe uno), y hacemos un for en el documento que creo previamente el C++, mientras reemplaza el carácter punto por comillas, y lo guardamos, de esta forma tenemos solucionado nuestro primer problema

Diapositiva #15

El segundo arreglador es básicamente el mismo funcionamiento, un for que reemplaza caracteres, pero en este caso es el guión bajo con un espacio.

Que esto pasó ya que usamos un string en el nombre de la calle y cuando teníamos que leer el archivo cada espacio nos lo tomaba con un fin de línea.

Ejemplo del error

Diapositiva #16

Este script lo llamamos compilador porque lo que hace es automatizar el trabajo de un usuario, primero le pusimos comandos de pura estética (como titulo y color verde en las letras) y creamos dos variables, el input que seria el Recorrido ya reparado por los dos scripts anteriores, y la salida sería la imagen del grafo, para esto llamamos a la utilidad Graphviz con las variables anteriormente mencionadas, y luego cerramos el script.

Diapositiva #17

Leer la diapositiva ( no leer lo azul)

Diapositiva #18

- Ejecutamos el programa