

**COMPLEJIDAD ALGORÍTMICA**

**Trabajo Final**

**SECCIÓN: WV72**

**PROFESOR: LUIS CANAVAL**

**Integrantes:**

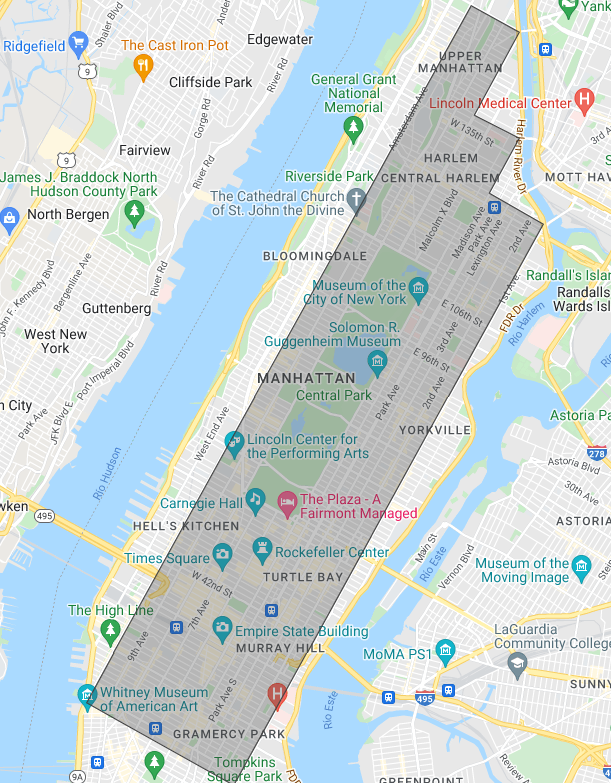
|  |  |  |
| --- | --- | --- |
| **APELLIDOS Y NOMBRES** | **CÓDIGOS** | **PARTICIPACIÓN** |
| San Martin Alvarez, Joaquin Alfonso | U201910511 | 100% |
| Romero Moran, William’s | U201816224 | 100% |
| Dioses Molina, Djalma | U201921405 | 100% |

**CICLO 2022-01**

1. **Resumen Ejecutivo**

Decidimos investigar y escoger una ciudad la cual no tenga calles que puedan ser curvas, redondeadas o deformes, esto con la finalidad de no tener errores o complicaciones para esta entrega, luego de un buen tiempo de buscar ciudades óptimas para este trabajo, optamos por usar una gran parte de la ciudad de New York, escogimos New York debido a su gran simetría en su terreno, permitiéndonos así la facilidad de realizar un grafo con un 95% de simetría a excepción de algunos lugares en donde no hay complicaciones.

1. **Imagen estática de la ciudad**

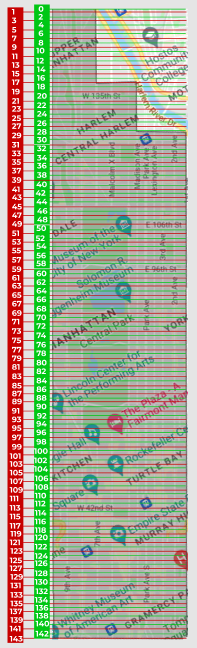


1. **Descripción de los datos consignados por calle**

Para el caso de las calles decidimos dividirlo en dividimos en 2 secciones:

**Calles Anchas**

En la siguiente imagen podemos visualizar las calles anchas más importantes, en donde tomamos en cuenta absolutamente todas las entradas que cuentan con un camino que va de izquierda a derecha o viceversa, finalmente contando las calles anchas tomadas nos da un total de 144. Puede visualizar el nombre de cada una de estas calles según su número asignado en el documento que hemos anexado en esta entrega.



**Calles Largas**

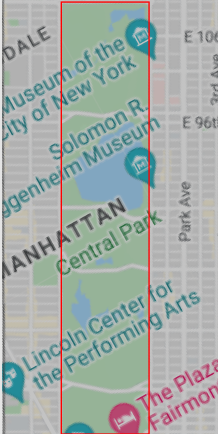
Como comentamos anteriormente, en la siguiente imagen podrá visualizar todas las calles largas que hemos seleccionado, para esta sección de calles tomamos en cuenta absolutamente todas las entradas que cuentan con camino que van de arriba hacia abajo o viceversa. Finalmente al contar las calles largas seleccionadas minuciosamente nos da una suma de 12 en donde podrá

visualizar el nombre de estas calles detalladamente en el archivo indexado para esta entrega.



1. **Descripción de la información consignada por intersección**

En el caso de las intersecciones entre calles pues al ser una ciudad simétrica en su mayor parte, no se nos complicó al realizar las intersecciones entre las calles anchas y largas, solo hay un pequeña excepción de complicación en las calles largas que van desde 2 hasta la 5 que interseccionan con las calles anchas que van desde la 45 hasta la 96, en este intervalo de intersecciones de calles anchas y largas no tienen conexión de intersecciones debido al central park.



1. **Explicación de cómo se elaboró el grafo, que representan las aristas y los vértices**

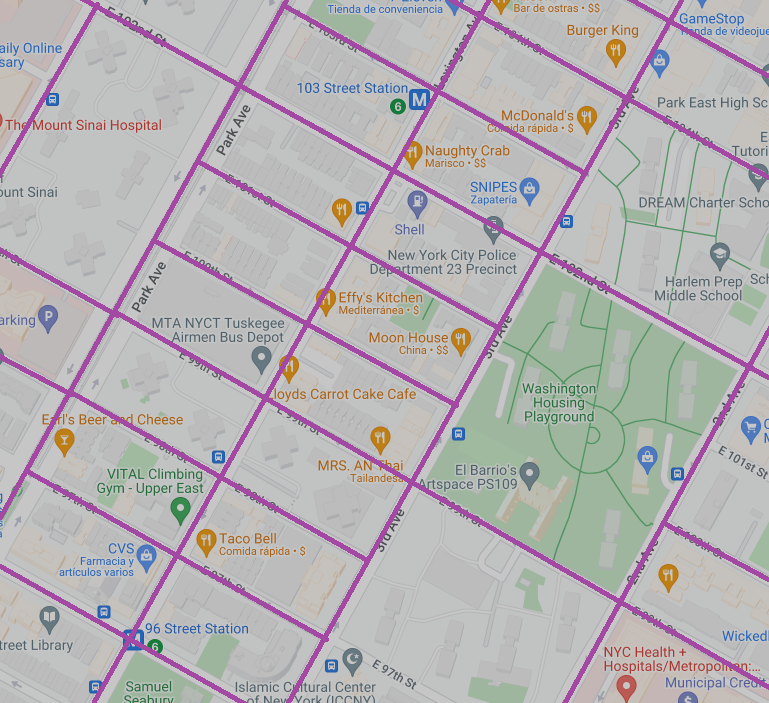
La ciudad al ser simétricamente perfecta, simplemente codificamos un pequeño programa en phyton

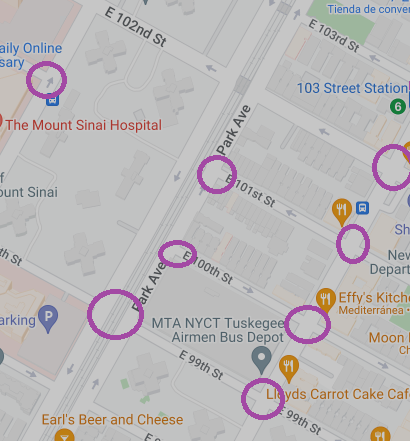


En la imagen anterior podemos observar el programa que crea el grafo de la ciudad en este caso invertimos el proceso de aristas en vez de unir esta vez quitamos aristas que no tengan conexión un vértice con otro, por otro lado se usó la biblioteca numpy para la generación de una matriz. Luego se implementó la función “removeEdgeMatriz” la cual nos permite cambiar los valores de la matriz. Por último se modificaron los valores con la ayuda de la estructura repetitiva “for” y la matriz generada se guardó dentro de un bloc de notas.

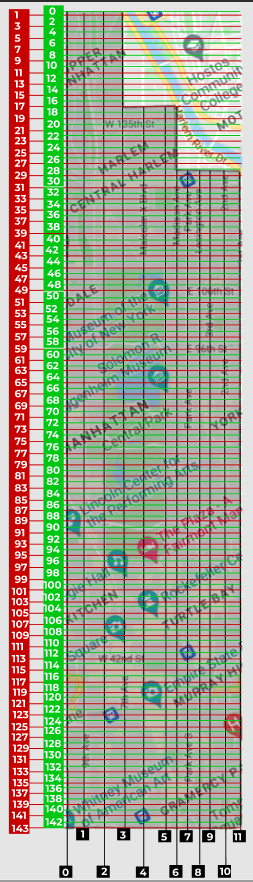
En el archivo indexado llamado matriz.txt puede visualizar el grafo generado.

En la siguiente imagen podemos observar que significan las aristas, están unen a los vértices que tiene conexión entre ellos, es decir que las aristas representan el recorrido que hay de una calle o cuadra a otra.



De la misma manera podemos observar en la siguiente imagen que los vértices, representan los paraderos o calles principales.

Finalmente una imagen de como quedo la ciudad con vértices y sus respectivas aristas implementadas:



1. **Código**

Durante el desarrollo del proyecto hemos necesitado crear algoritmos necesarios para identificar rutas cortas con respecto a dos puntos aleatorios. Uno de los algoritmos que implementamos es Dijkstra, el cual nos permite identificar el camino más corto, dado un vértice origen y un vértice final.

Además, hemos utilizado la herramienta de Miniconda para poder crear una página en localhost la cual nos permite mostrar el grafo de la ciudad seleccionada. Esta página utiliza tecnologías como Python, Css, Javascript y Html, los cuales nos ayudan a mostrar lo necesario en dicha página, por ejemplo, los caminos hallados por el algoritmo de Dijkstra y la lista de adyacencia generada por nosotros

1. **Conclusiones**

* Desde el inicio de nuestro proyecto, nos hemos dado cuenta de que el trabajo en equipo es muy importante, pues siempre es necesaria la comunicación constante entre cada uno de los miembros del equipo y el uso de herramientas que nos ayuden a avanzar de manera progresiva en un trabajo coordinado y fluido.
* Siempre es necesario apoyarse al momento de tener dudas, pues siempre es necesario aprender algo nuevo para mejorar la eficiencia en el desarrollo del proyecto.
* La comunicación constante es necesaria, siempre utilizando las herramientas que nos permitan facilitar dicho proceso, como Discord. Además, también es muy importante conocer herramientas como Github, Visual Studio Code o Google Colab.

**Anexos:**

Exposición del trabajo parcial: <https://youtu.be/u3J01bcFNTQ>

Exposición del trabajo final: