

Complejidad Algorítmica

Trabajo Final

Integrantes

Daniel Jiménez

Diego Alosilla

Marco de la Cruz

Contenido

[Introducción 1](#_Toc12566965)

[Motivación 2](#_Toc12566966)

[Objetivos 2](#_Toc12566967)

[Marco Teórico 2](#_Toc12566968)

[1. UDFS 2](#_Toc12566969)

[2. PRIM 2](#_Toc12566970)

[3. BELLMAN-FORD 2](#_Toc12566971)

[Análisis de Complejidad 2](#_Toc12566972)

[Algoritmo 01 (UDFS) 2](#_Toc12566973)

[Algoritmo 02 (BELLMANFORD) 3](#_Toc12566974)

[Algoritmo 03 (PRIM) 3](#_Toc12566975)

[Diseño de experimentos 3](#_Toc12566976)

[Diseño de plan de pruebas 3](#_Toc12566977)

[Diseño de casos de pruebas 3](#_Toc12566978)

[Desarrollo de experimentos 4](#_Toc12566979)

[Pruebas 4](#_Toc12566980)

[Análisis e interpretación de datos 8](#_Toc12566981)

[Resultados de la prueba 8](#_Toc12566982)

[Algoritmo 01 (UDFS) 8](#_Toc12566983)

[Algoritmo 02 (BELLMANFORD) 8](#_Toc12566984)

[Algoritmo 03 (PRIM) 8](#_Toc12566985)

[Conclusiones 9](#_Toc12566986)

[Bibliografía 10](#_Toc12566987)

# Introducción

El problema del agente viajero (tsp) es uno de los más estudiados en el campo de la optimización, ya que cuenta con diversas aplicaciones en la industria. Hace referencia a la visita lugares o nodos (para entregar o recoger mercancías) con el fin resolver problemas que impidan minimizar algún objetivo (tiempo o costos) o maximizar algún otro.

La primera solución reportada para resolver el problema del Agente Viajero fue en 1954, cuando George Dantzig, Ray Fulkerson, y Selmer Johnson publicaron la descripción de un método de solución del PAV (Problema del Agente Viaje o sus siglas en inglés TSP – Travel Sailsman Problem) titulado “Solutions of a large scale traveling salesman problem“ (Soluciones de gran escala para el problema del agente viajero) para resolver una instancia de 49 ciudades donde un agente viajero desea visitar un conjunto de ciudades, asignándoles un costo por visitar ciudades contiguas (distancia de traslado entre dos ciudades). Para esta solución se propusieron 2 condiciones: regresar a la misma ciudad de la cual partió y no repetir ciudades con el objetivo de encontrar una ruta o un camino con el menor costo posible.

# Motivación

Resolver el problema del agente viajero (TSP) ya que es uno de los más estudiados en el campo de la optimización, lo tomamos como un reto, de esta forma nos motiva a investigar sobre los algoritmos más convenientes para este problema.

## Objetivos

* Implementar un algoritmo el cual resuelva de manera eficiente el problema de las conexiones de los centros poblados en el país.
* Analizar la conveniencia de usar las técnicas de "UDFS", "MST", "BellmanFord", "Jhonson", "Floyd-Warshall"
* Analizar la complejidad del algoritmo que se implementará.

# Marco Teórico

Para la implementación del trabajo hemos utilizado el lenguaje de programación con los siguientes algoritmos:

## 1. UDFS

* El algoritmo UFDS es un algoritmo eficiente y rápido con pocos datos ,pero si se usan cantidad masiva de datos se demorara un tiempo muy excesivo, ya que compara varios arboles(rutas) que tomara en llegar de un punto a otro. Como también la suma de la distancia final, lo cual al final tomara la menor ruta utilizada de los arboles.

## 2. PRIM

* El algoritmo incrementa continuamente el tamaño de un árbol, comenzando por un vértice inicial al que se le van agregando sucesivamente vértices cuya distancia a los anteriores es mínima. Esto significa que en cada paso, las aristas a considerar son aquellas que inciden en vértices que ya pertenecen al árbol.

## 3. BELLMAN-FORD

* Un algoritmo que busca el camino más corto de un nodo a otro.
* Recibe como dato de entrada un Grafo y el nodo de salida. El algoritmo buscará el camino más corto desde el nodo de partida y empezará a buscar el camino más camino corto comparando el costo de cada uno.

## Análisis de Complejidad

### Algoritmo 01 (UDFS)

O(n!)

### Algoritmo 02 (BELLMANFORD)

O(|V|\*|E|)

### Algoritmo 03 (PRIM)

O(n^2)

# Diseño de experimentos

## Diseño de plan de pruebas

1. En el primer dataset existen 145’226 centros poblados, con un atributo que permite diferenciar si un centro poblado es capital regional, provincial, distrital o ninguno:

2. En el segundo dataset existen 75’513 centros educativos:

## Diseño de casos de pruebas

En el dataset 1:

Aplicar su solución a las 25 capitales regionales.

Aplicar su solución a las 171 capitales provinciales.

Aplicar su solución a las 1’678 capitales distritales.

Aplicar su solución a los 143’351 centros poblados restantes.

En el dataset 2:

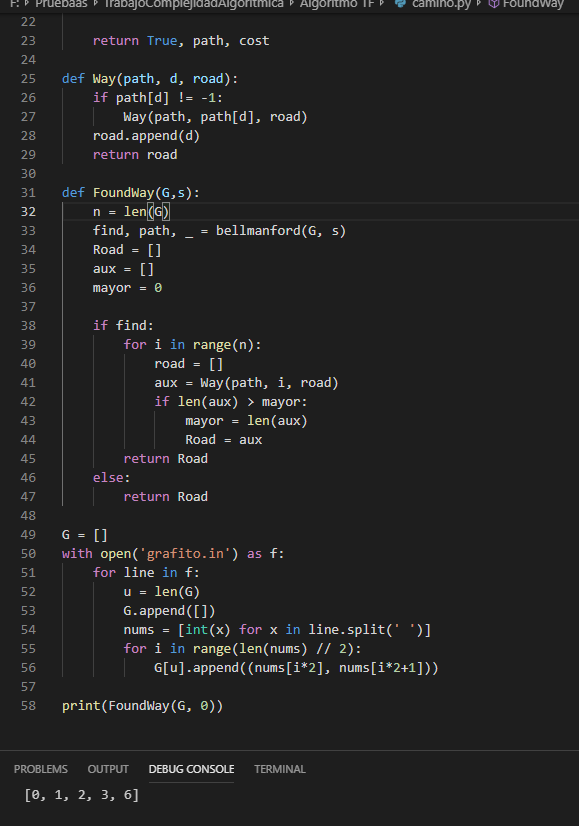
Aplicar su solución a los centros educativos del distrito donde vive en Lima.

Aplicar su solución a los centros educativos de la región y ciudad de Lima.

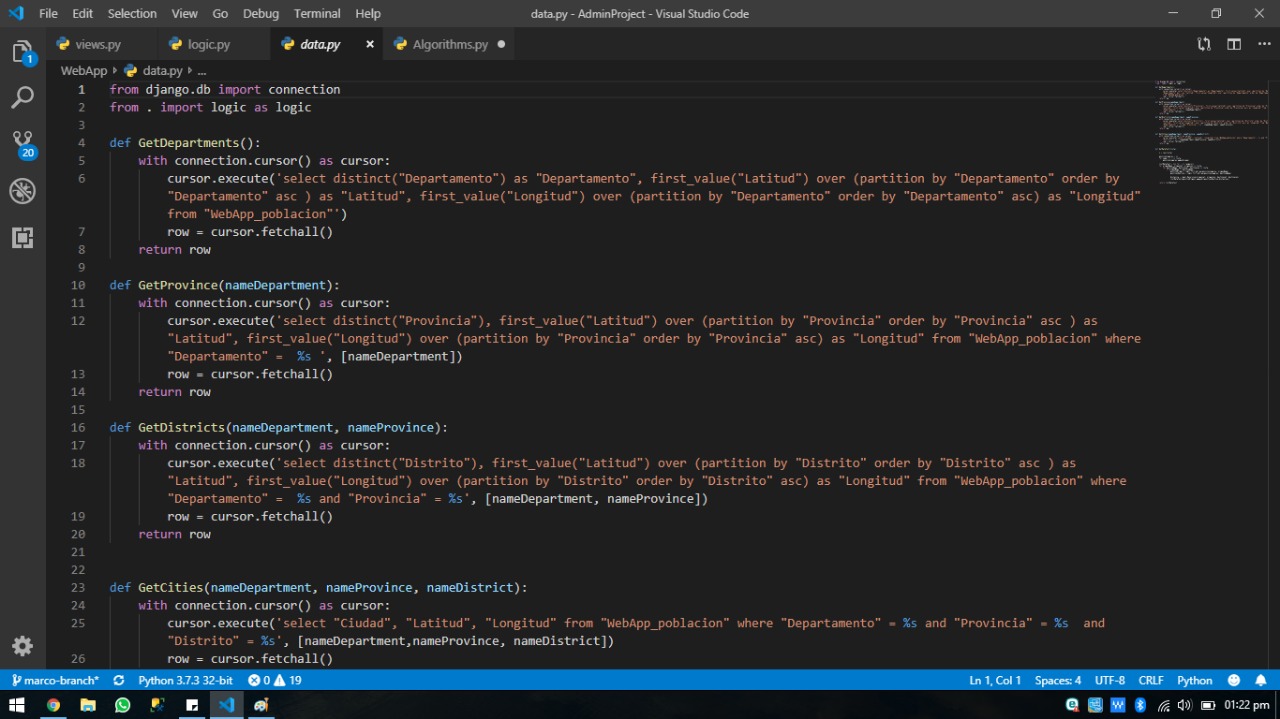
Aplicar su solución a los 75’513 centros educativos

# Desarrollo de experimentos

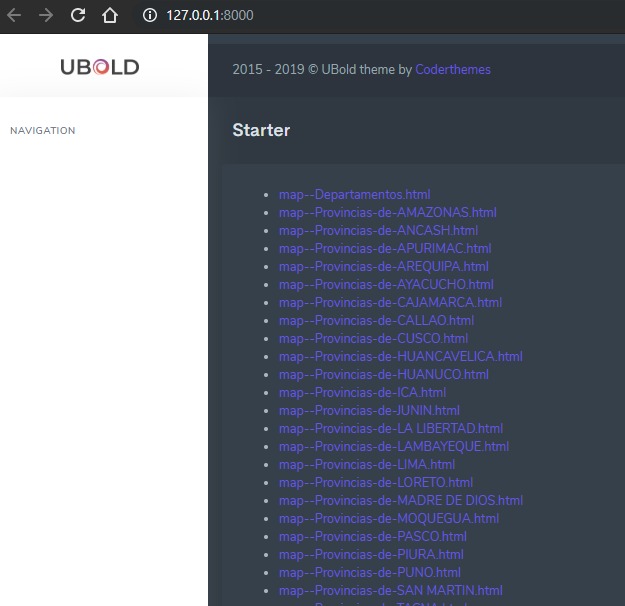
## Pruebas



Solución de algoritmo



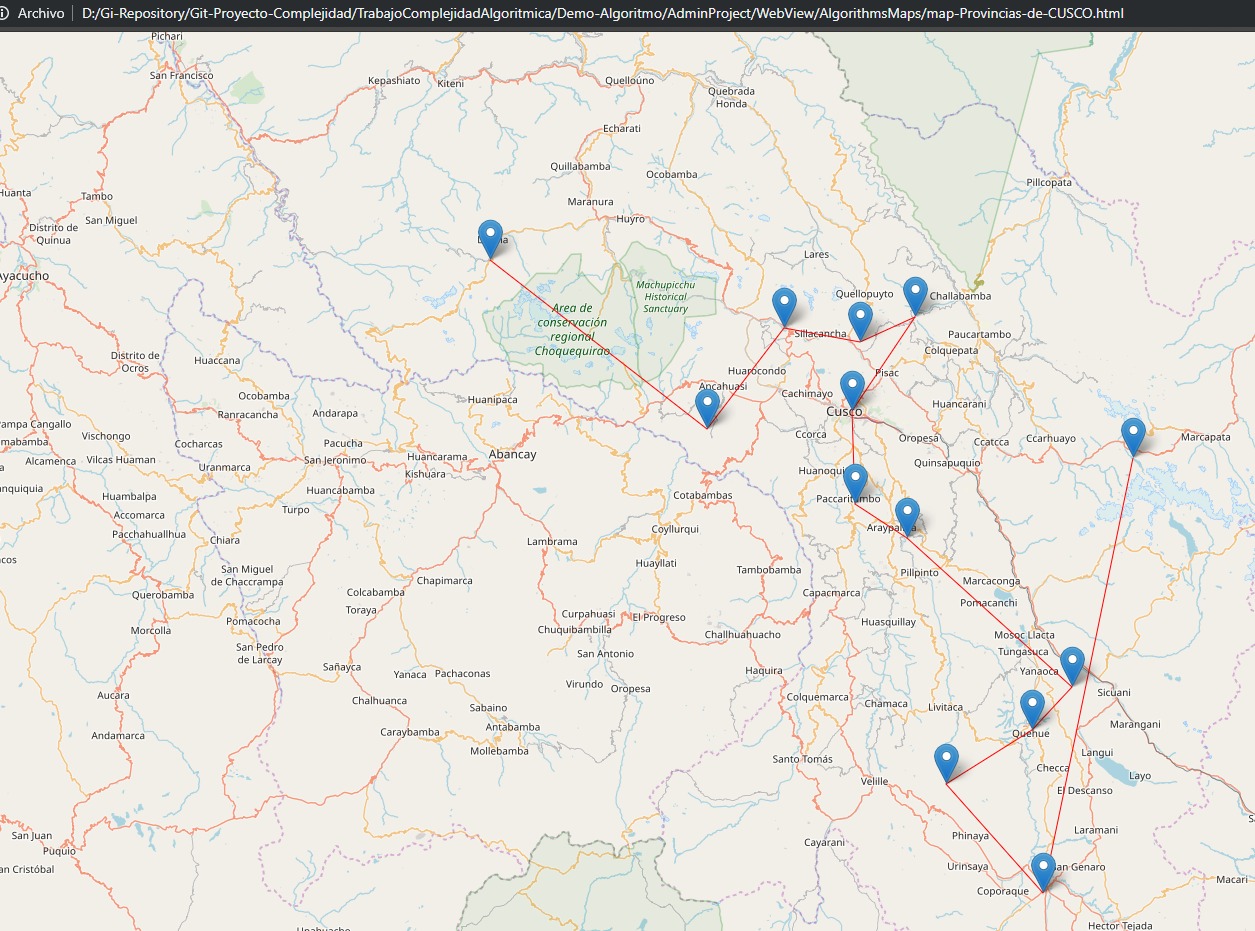
Los datos jalados del dataset para su uso.



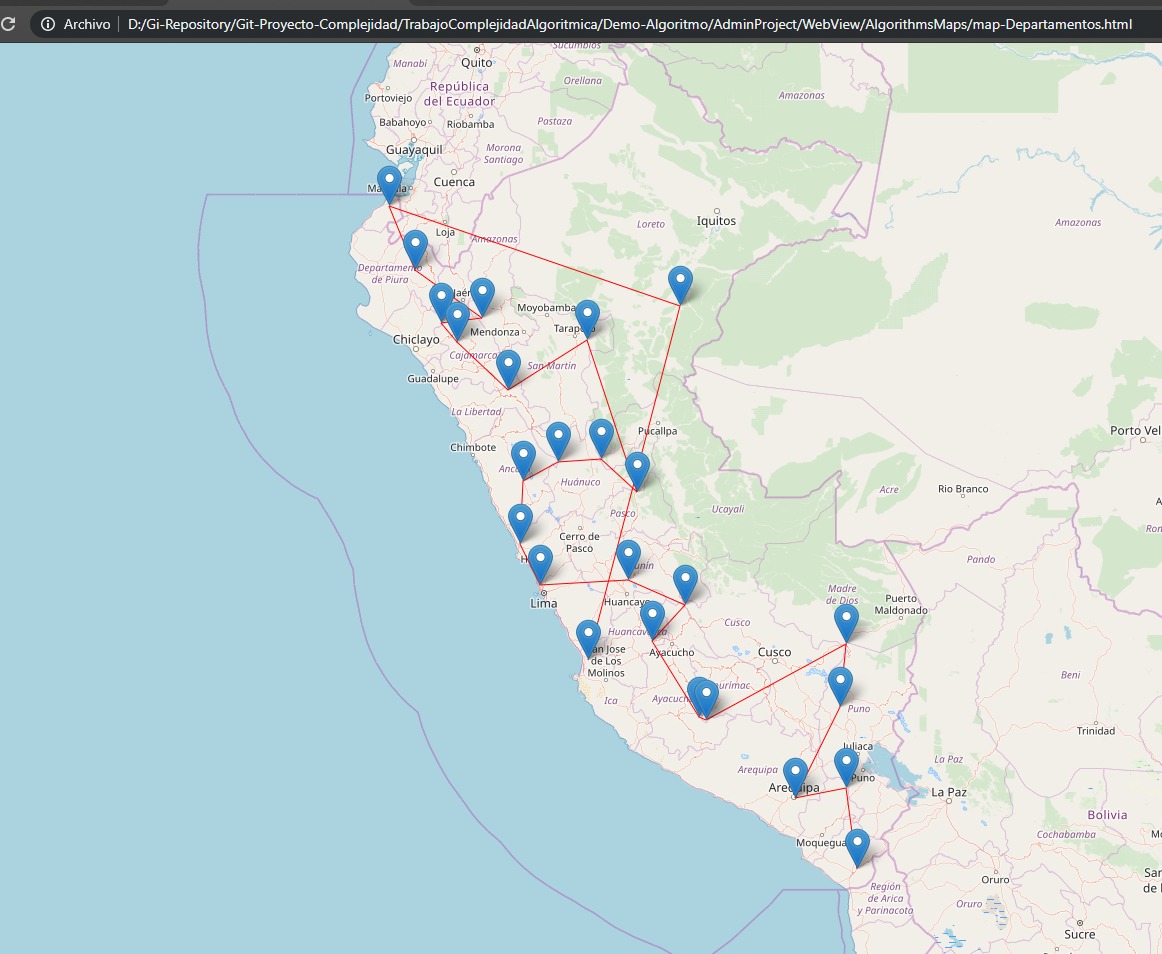
Visualización de los datos



Crea el archivo del mapa



Prueba 01



Prueba 02

# Análisis e interpretación de datos

## Resultados de la prueba

### Algoritmo 01 (UDFS)

Al tener una complejidad de n!, el tiempo de ejecución es altísimo por lo que los datos al momento de procesarlos demoran en encontrar una solución.

### Algoritmo 02 (BELLMANFORD)

El algoritmo de Bellmanford usado para resolver el problema posee una complejidad |V|\*|E| en el peor de los casos por lo que el tiempo de ejecución del algoritmo para una cantidad de datos grande podría demorar mucho tiempo. Por ejemplo, para calcular el camino en un grafo con 7 nodos se demora un tiempo muy pequeño, casi los 0.00234 milisegundos pero para mas de 1200 datos puede llegar hasta las 58 minutos.

### Algoritmo 03 (PRIM)

El algoritmo de Prim tiene una complejidad n^2 por lo que su tiempo de ejecucion tendrá un crecimiento bastante acelerado. Con 2000 datos se puede llegar a tomar el tiempo de 1.11 horas.

# Conclusiones

* El programa no llega a cargar todos los datos del archivo csv de donde se extrae la información.
* Resulta difícil hacer que uno de los algoritmos implementados retorne el camino más corto, puesto que al crear las conexiones entre los centros poblados se hace de forma aleatoria y el volumen de datos aumenta significativamente.
* El problema del Agente Viajero (TSP) es un problema cuya solución ha sido estudiada desde los inicios de la Inteligencia Artificial considerando que su aplicación puede ser en cualquier área de estudio cuyos problemas reflejen una situación donde se tienen diferentes puntos a visitar con un costo considerado en el enlace entre dichos puntos (costo: recursos empleados como distancia, tiempo, monto económico, etc.). Cada autor ha propuesto soluciones para ciertas instancias de TSP, cada uno con una perspectiva diferente empleando técnicas que no son repetibles pero que, en determinado momento, se pueden emplear para dar lugar a nuevas soluciones; de las técnicas empleadas la más común es el uso de redes neuronales dada su similitud, donde cada neurona es un nodo a visitar y las relaciones entre neuronas es el vector que representa el costo.

# Bibliografía

https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n3/e5.html

http://editorial.ucentral.edu.co/ojs\_uc/index.php/Ingeciencia/article/download/310/277

https://networkx.github.io/documentation/networkx-2.3/tutorial.html