

UBER.py

Proyecto semestral

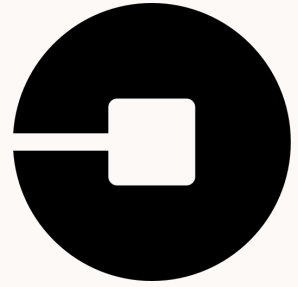


Tomás Rando y Manuel Quesada



<https://github.com/Kilxz/ProyectoAlgo2>

Problemas encontrados



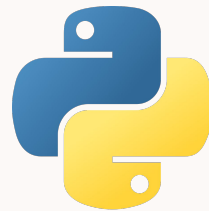
- Manejo de argumentos
- Dijkstra
- Primera solución muy compleja
- Casos al calcular las distancias
 - Cuando la persona estaba en la misma calle que algún auto
 - Cuando la persona o el auto estaban en la mitad de la calle (Caso promedio)
- Cálculo de las distancias de los autos a las ubicaciones
- Número de esquinas no en orden

Soluciones



- Se uso la libreria sys para el uso de argumentos
- Dijkstra
 - Primera versión: $O(V^3)$
 - Segunda versión: $O(V^2 \cdot \log V)$
 - Versión actual: $O(A \cdot \log V)$
- Primera solución: Cargar personas y autos en el mismo grafo e irlos moviendo con cada inserción
Solución actual: Cargar personas y autos en un hash aparte y trabajar con las distancias a las esquinas
- Caso 1: Se re-utilizó una lista con esquinas a la que se le insertaba los autos que estaban en una calle específica.
Caso 2: Utilizando las distancias esquina-persona y esquina-autos se fueron sumando a la hora de agregarlas para obtener el costo verdadero
- Se precalcula las distancias de los autos a las esquinas, añadiendolas a una priorityQueue asociada a cada esquina
- A cada esquina se le da un vértice asociado que se guarda en una hash para su posterior acceso

Posibles mejoras



Diferente implementación
de PriorityQueue en insert
de autos

Insertar las
ubicaciones móviles
dentro del grafo

Diferente manejo del
ranking de autos en
calles doble mano

Complejidad espacial



- **3 HASH (longitud m)**
 - HASH DE ESQUINAS
 - HASH DE PERSONAS Y AUTOS
 - HASH DE UBICACIONES FIJAS
- **LISTAS (longitud m)**
 - LISTA DE ESQUINAS CON LISTAS DE PRIORIDAD
 - LISTA DE ESQUINAS CON VÉRTICE ASOCIADO
- **1 GRAFO REPRESENTADO POR LISTAS DE ADYACENCIA**

SIENDO m LA CANTIDAD DE ESQUINAS

Complejidad Temporal

Inserción

$O(n^2)$

Viaje

$O(n^2)$

CREATE MAP

```
graph TD; A[CREATE MAP] --> B[Traducción del mapa]; A --> C[Componer mapa (lista de adyacencia)]; A --> D[Creacion de HASH TABLES];
```

Traducción del mapa

Componer mapa (lista de adyacencia)

Creacion de HASH TABLES

LOAD MOVIL
ELEMENT

Autos:

Se inserta el auto a la hash table de
ubicaciones móviles

Se realiza un dijkstra que almacena en
cada una de las esquinas una lista de
prioridad una tupla del auto y el precio a
pagar

Personas:

Se inserta a la hash table de ubicaciones
móviles

LOAD FIX ELEMENT

Insercion en el grafo mediante vértice asociado

Insercion en Hash Table de ubicaciones fijas

CREATE TRIP

Dijkstra desde la persona para saber si es posible llegar a la dirección

Re-utilizando el dijkstra, se obtiene el camino más corto a la dirección

Se imprime el ranking de los 3 autos mas cercanos a la persona, en cuanto a direccion y costo

Realizada la elección, realiza el movimiento

Se modifica el hash de ubicaciones móviles y las listas de prioridad

HASH UBICACIONES MÓVILES

0	—	"P1"	("P1", ("e1", 5, "e2", 24), 5000)
1			
2			
3	—	"C5"	("C5", ("e2", 10, "e3", 5), 200)
4			

HASH UBICACIONES FIJAS

0				
1	————	<table><tr><td>"H3"</td><td>("H3", 5, ("e1", 25, "e2", 4))</td></tr></table>	"H3"	("H3", 5, ("e1", 25, "e2", 4))
"H3"	("H3", 5, ("e1", 25, "e2", 4))			
2	————	<table><tr><td>"H1"</td><td>("H1", 6, ("e2", 10, "e3", 5))</td></tr></table>	"H1"	("H1", 6, ("e2", 10, "e3", 5))
"H1"	("H1", 6, ("e2", 10, "e3", 5))			
3				
4				

The background features abstract organic shapes and lines. On the left, a large light beige circle is partially visible, with a dark brown wavy line curving around its edge. On the right, a similar dark brown wavy line curves around a light beige shape. The central area is a plain light beige color.

Gracias