

Integrantes grupo 6:

- Sebastian Heredia Vargas | s.herediav@uniandes.edu.co | 202012346
- Felix Samuel Rojas Casadiego | fs.rojas@uniandes.edu.co | 202013414

Pregunta 1: ¿Qué características tiene el grafo definido?, ¿Tamaño inicial, es dirigido?, ¿Estructura de datos utilizada?

El grafo está compuesto Tablas de hash para guardar los vértices del grafo, grafos para representar las rutas entre estaciones, información de los componentes conectados y los caminos de costo mínimo desde un vértice determinado a todos los otros vértices del grafo.

El tamaño inicial está definido como 14000.

El grafo se define como dirigido en la línea `directed=True`.

Se utiliza la estructura de datos `ADJ_LIST` que corresponde a las listas de adyacencias.

Pregunta 2: ¿Qué instrucción se usa para cambiar el límite de recursión de Python? ¿Por qué considera que se debe hacer este cambio?, ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

Para cambiar el límite de recursión de Python se usa la instrucción:

```
sys.setrecursionlimit(recursionLimit)
```

Este cambio se usa para maximizar el número de usos de una función recursiva.

Usualmente Python pone por defecto 1000 como numero limite.

Pregunta 3: ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4? (Ayuda: ¿es un crecimiento lineal?)

Tamaño	Número de vértices	Número de arcos	Tiempo de ejecución (Opción 4)
50	74	73	0,0316985
150	146	146	0,0452832
300	295	382	0,1006631
1000	984	1633	0,4147311
2000	1954	3560	1,7077299
3000	2922	5773	2,76470100
7000	6829	15342	7,13041270
10000	9767	22768	28,26631140
14000	13535	32301	41,29856960

La relación que existe entre el número de vértices, arcos y el tiempo que toma la operación 4 es un crecimiento de la forma $O(n * \log(n))$ o linealítmico con respecto al número de arcos, ya que entre más arcos haya mayor será el número de posibles rutas entre estaciones (vértices) que se deben visitar hasta encontrar la ruta más corta para todas las estaciones.