

OBSERVACIONES DEL LA PRACTICA

Sebastián Contreras Salazar Cod 202020903

Valentina Goyeneche cod 201922380

a) ¿Qué instrucción se usa para cambiar el límite de recursión de Python?

La instrucción para cambiar el limite de recursión de Python es `sys.setrecursionlimit(n)` en el que es posible modificar el tamaño maximo de la pila de recursión, en el que n es el limite de la pila de ejecución.

b) ¿Por qué considera que se debe hacer este cambio?

Es necesario hacer este cambio, puesto que así se puede evitar llenar la memoria de los computadores o permitir que el programa consuma más memoria, debido a que cada llamada recursiva agrega un elemento a la pila de llamadas a las funciones y a la memoria, por lo que al hacer este cambio se puede limitar el uso de la memoria y poder permitir que el programador pueda “manipular” o controlar tambien el consumo de la memoria.

c) ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

El limite por omisión es de 1000 llamadas recursivas.

d) ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4?

El número de vértices, el número de arcos del grafo, y el tiempo que toma esta instrucción con cada uno de los archivos CSV son los siguientes:

Para el csv de 50 se tiene que hay un numero de arcos 73, un numero de vertices de 74 y un tiempo de 219.80 para la opción 4.

```
48 | servicefile = 'bus_routes_50.csv'
49 | initialStation = None
50 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

Numero de arcos: 73
El limite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>7
Estación: 96289 Total rutas servidas: 1

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
219.80327099999886
```

Para el csv de 150 se tiene que hay un numero de arcos 146, un numero de vertices de 146 y un tiempo de 194.0 para la opción 4.

```
48 | servicefile = 'bus_routes_150.csv'
49 | initialStation = None
50 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>

Cargando información de transporte de singapur ....
Numero de vertices: 146
Numero de arcos: 146
El limite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
194.008810000003
```

Para el csv de 300 se tiene que hay un numero de arcos 382, un numero de vertices de 295 y un tiempo de 233.9 para la opción 4.

```
48 | servicefile = 'bus_routes_300.csv'
49 | initialStation = None
50 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>2

Cargando información de transporte de singapur ....
Numero de vertices: 295
Numero de arcos: 382
El limite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
233.9362399999991
```

Para el csv de 1000 se tiene que hay un numero de arcos 1633, un numero de vertices de 984 y un tiempo de 2356.8 para la opción 4.

```
48 | servicefile = 'bus_routes_1000.csv'
49 | initialStation = None
50 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>2

Cargando información de transporte de singapur ....
Numero de vertices: 984
Numero de arcos: 1633
El limite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
2356.082064000001
```

Para el csv de 2000 se tiene que hay un numero de arcos 3560, un numero de vertices de 1954 y un tiempo de 6749.361 para la opción 4.

```
48 | servicefile = 'bus_routes_2000.csv'
49 | initialStation = None

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>
Cargando información de transporte de singapur ....
Numero de vertices: 1954
Numero de arcos: 3560
El limite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
6749.361968000001
```

Para el csv de 3000 se tiene que hay un numero de arcos 5773, un numero de vertices de 2992 y un tiempo de 12272.65 para la opción 4.

```
48 | servicefile = 'bus_routes_3000.csv'
49 | initialStation = None
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estación base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>2

Cargando información de transporte de singapur ....
Numero de vertices: 2922
Numero de arcos: 5773
El límite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estación base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
12272.65129
```

Para el csv de 7000 se tiene que hay un numero de arcos 15334, un numero de vertices de 6829 y un tiempo de 57139.38 para la opción 4.

```
48 | servicefile = 'bus_routes_7000.csv'
49 | initialStation = None

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>

Cargando información de transporte de singapur ....
Numero de vertices: 6829
Numero de arcos: 15334
El límite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
57139.388281
```

Para el csv de 10000 se tiene que hay un numero de arcos 22758, un numero de vertices de 9767 un tiempo de 163385.54 para la opción 4.

```
48 | servicefile = 'bus_routes_10000.csv'
49 | initialStation = None

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

*****
Seleccione una opción para continuar
>

Cargando información de transporte de singapur ....
Numero de vertices: 9767
Numero de arcos: 22758
El límite de recursion actual: 1048576

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
>4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
163385.549936
```

Para el csv de 14000 se tiene que hay un numero de arcos 32270, un numero de vertices de 13535 y un tiempo de 379117.9 para la opción 4

```
47
48 servicefile = 'bus_routes_14000.csv'
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: f
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
~2
Cargando información de transporte de singapur ....
Numero de vertices: 13535
Numero de arcos: 32270
El limite de recursión actual: 1048576
*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
0- Salir
*****
Seleccione una opción para continuar
~4
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
269046.610168
```

Ahora bien, la relación que cremos que existe entre el numero de vertices, arcos y tiempos que toma la operación 4 es que a mediad que hay mayores arcos y vertices, debe tomar mayor tiempo la operación 4, puesto que al calcular los caminos de costo minimo desde la estación initalStation a todos los demas vertices del grafo, el tiempo en lograr esta función aumentaria si hay una mayor cantidad de arcos y vertices, debido a que aumentaria el catalogo y aumentaria el tiempo en lograr esa operación. Del mismo modo, se puede observar que a mediada que hay un mayor numero de arcos, aumenta tambien el numero de vertices, sin embargo, aumenta considerablemente más el numero de arcos al numero de vertices.

e) ¿Qué características tiene el grafo definido?

Las características que tiene el grado definido es que esta tiene una dirección para los arcos entre los vertices, puesto que los arcos estan dirigidos, en donde teoricamente se verian con flechas, en el que los datos deben ser especificos. En otras palabras, un grafo definido cuenta con una dirección de ida para los arcos entre los vertices, en el que solo puede ir a un lado, mas no puede tener una idea y vuelta como en los grafos no definidos.

f) ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo es de 14000, puesto que como se puede ver en el grafo de connections, el size es de 14000 datos.

```
analyzer = {
    'stops': None,
    'connections': None,
    'components': None,
    'paths': None
}

analyzer['stops'] = m.newMap(numelements=14000,
                             maptype='PROBING',
                             comparefunction=compareStopIds)

analyzer['connections'] = gr.newGraph(datastructure='ADJ_LIST',
                                       directed=True,
                                       size=14000,
                                       comparefunction=compareStopIds)
```

g) ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada es de un mapa en la llave de “stops” y un grafo en la llave de “connections”, siguiendo lo que se ve en la imagen de arriba

h) ¿Cuál es la función de comparación utilizada?

La función de comparación utilizada tanto para el mapa como para el grafo es de compareStopIds, que básicamente es como los otros compares que hemos trabajado a lo largo del semestre, con la diferencia de que se comparan dos estaciones al mirar la llave y una parada. El código de la comparación es el siguiente:

```
def compareStopIds(stop, keyvaluestop):
    """
    Compara dos estaciones
    """
    stopcode = keyvaluestop['key']
    if (stop == stopcode):
        return 0
    elif (stop > stopcode):
        return 1
    else:
        return -1
```