

OBSERVACIONES DE LA PRÁCTICA

Pablo Pedreros - Código: 202112491

Preguntas de análisis

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

La instrucción que se usa es “`sys.setrecursionlimit(n)`”, siendo n el límite de veces que una función recursiva puede llamarse a sí misma, en este caso el límite está puesto en 2^{20} .

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

Si el límite de recursión no se cambia, sino que se deja el predeterminado de Python y este es menor que el número de recursiones que necesita nuestro programa, Python generará un error cuando una función recursiva (función que se llama a sí misma) trate de llamarse a sí misma más veces que el límite. Esto pasaría por ejemplo en la tercera opción del menú si dejáramos el límite de recursión como el predeterminado de Python.

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

Tiene 1000 como límite de recursión.

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

Número de vértices	Número de arcos	Tiempo de ejecución (ms)
74	73	15.625
146	146	15.625
295	382	46.875
984	1633	234.375
1954	3560	828.125
2922	5773	1484.375
6829	15334	3562.5
9767	22758	12031.25
13535	32270	21640.625

El número de arcos y el tiempo de ejecución van aumentando exponencialmente mediante aumentan el número de paradas a considerar, pues entre más rutas haya para entreconectar, más y más arcos habrá que conecten las nuevas rutas con las rutas que ya estaban mediante nuevos arcos de un peso n para conectar las estaciones de las nuevas rutas, y nuevos arcos de peso 0 para conectar las estaciones de las rutas que ya había con las mismas estaciones pero de las nuevas rutas. Como la función busca el camino más corto del vértice base que recibe a todas las otras estaciones (o vértices) del grafo, entre más vértices haya, más aumentará el tiempo de búsqueda, pues más posibles caminos tendrá el algoritmo para buscar el más corto.

e) ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?

Se podría decir que el grafo es disperso, pues en general las estaciones no se relacionan todas con todas, ni mucho menos, sino que se relacionan solo con las mismas estaciones, sino que de otra ruta y con las

dos estaciones adyacentes de su misma ruta, así que comparando el número de relaciones por vértice con el número total de vértices, podemos decir que es bastante disperso. Podemos confirmar esto dividiendo el número de arcos sobre el número de arcos posibles y veremos que el resultado no llega ni cerca a 0.1, por lo que hablamos de un grafo muy disperso.

El grafo será dirigido, en el archivo CSV se especifica en qué dirección se hace cada servicio y hay varias rutas que se hacen en una sola dirección, por lo que el grafo conectará las estaciones con una estación de salida y una de llegada, no con arcos no dirigidos (cosa que podemos ver desde las líneas en las que se implementa el grafo).

No, el grafo no está fuertemente conectado, pues el programa nos dice que hay 30 componentes conectados, lo que significa que hay 30 componentes separados dentro del grafo y que no es posible que este sea fuertemente conexo, pues no habrá forma de que un vértice de un componente llegue a relacionarse con un segundo vértice de otro componente.

f) ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo en la implementación es de 14000

g) ¿Cuál es la Estructura de datos utilizada?

EL grafo implementado funciona por medio de listas de adyacencia, hayando un balance entre espacio en memoria y tiempo de las funciones.

h) ¿Cuál es la función de comparación utilizada?

Es la función “comparestopsid”, que compara dos estaciones para ver si son iguales en código de estación y número de ruta a la hora de crear los vértices dentro del grafo.

*Para la función 6:

Número de vértices	Número de arcos	Tiempo de ejecución (ms)
74	73	0
146	146	0
295	382	0
984	1633	0
1954	3560	0
2922	5773	15.625
6829	15334	15.625
9767	22758	15.625
13535	32270	15.625