

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

Es un grafo dirigido, con un tamaño inicial de 14000, correspondiente al número de vértices esperados. Se hace uso de una lista 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?

La instrucción que se usa para modificar el límite de recursión está en el archivo view:

sys.setrecursionlimit(recursionLimit)

El parámetro *recursionLimit* corresponde al límite de recursión establecido por nosotros, que en este caso es 20000.

Considero que este cambio se debe hacer debido al uso de algoritmos recursivos como es el caso del DFS. Al ingresar un valor más grande, permitimos una mayor capacidad en la Pila de ejecución, y con esto, facilitamos que el máximo número de recursiones que se permiten en una función recursiva sea mayor.

El valor inicial que tiene Python como límite de recursión es 1000.

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?)

Es bastante curioso, si se observan los tiempos de los archivos (en segundos): bus_routes_1000.csv, bus_routes_2000.csv, bus_routes_3000.csv; donde la diferencia entre el número de vértices es de aproximadamente 1000, se podría creer que:

Tiempo de ejecución bus_routes_1000.csv = 0,3

Tiempo de ejecución bus_routes_2000.csv = 0,9 = $0,3 * 1(3)$

Tiempo de ejecución bus_routes_3000.csv = 1,8 = $0,3 * 2(3)$

Siendo así, el tiempo de ejecución para bus_routes_7000.csv, estaría dado por la fórmula:

Tiempo de ejecución bus_routes_7000.csv = $8,4 \neq 0,3 \cdot 6(3)$

Pero esta igualdad no se cumple. Esto nos lleva a creer que el número de arcos es fundamental si quisiéramos hallar una fórmula exacta que nos determine la relación entre tiempo de ejecución, número de arcos y número de vértices.

De igual manera, por la aparente relación que tienen los tiempos de ejecución de los archivos con 1000, 2000 y 3000 datos, podríamos creer que realmente si se presenta un crecimiento lineal.

Archivo: bus_routes_50.csv

Número de Vértices: 74.

Número de Arcos: 73.

Tiempo de ejecución:
0.033042499999993424.

Archivo: bus_routes_150.csv

Número de Vértices: 146.

Número de Arcos: 146.

Tiempo de ejecución:
0.044411800000002444.

Archivo: bus_routes_300.csv

Número de Vértices: 295.

Número de Arcos: 382.

Tiempo de ejecución:
0.077693499999999875.

Archivo: bus_routes_1000.csv

Número de Vértices: 984.

Número de Arcos: 1633.

Tiempo de ejecución: 0.3618641000000018.

Archivo: bus_routes_2000.csv

Número de Vértices: 1954.

Número de Arcos: 3560.

Tiempo de ejecución: 0.95600089999999994.

Archivo: bus_routes_3000.csv

Número de Vértices: 2922.

Número de Arcos: 5773.

Tiempo de ejecución: 1.98324869999999972.

Archivo: bus_routes_7000.csv

Número de Vértices: 6829.

Número de Arcos: 15342.

Tiempo de ejecución: 8.4634426

Archivo: bus_routes_10000.csv

Número de Vértices: 9767.

Número de Arcos: 22768.

Tiempo de ejecución: 19.2750871999999998.

Archivo: bus_routes_14000.csv

Número de Vértices:

Número de Arcos:

Tiempo de ejecución: 27.6187245