

LABORATORIO No. 9: Grafos

Valentina Perea Marquez – 202013095

Mateo Lopez Cespedes - 202014481

Preguntas

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

Se usa la función `sys.setrecursionlimit()`. Su parámetro será un entero que determinará la profundidad máxima

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

Cuando el número de iteraciones sobre un conjunto de datos supera el límite de recursión, será necesario aumentar dicho límite

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

Por defecto Python maneja un límite de recursión de 1000

TABLA DE REGISTRO OPCION CUATRO POR TAMANO DE DOCUMENTO CSV

Registre número de vértices, el número de arcos del grafo, y el tiempo que toma esta instrucción con cada uno de los archivos CSV en el documento de observaciones del laboratorio

DOCUMENTO	NUMERO DE VERTICES	NUMERO DE ARCOS	TIEMPO
bus_routes_50	74	73	0.03125
bus_routes_150	146	146	0.046875
bus_routes_300	295	382	0.0625
bus_routes_1000	984	1633	0.28125
bus_routes_2000	1954	3560	1.15625
bus_routes_3000	2922	5773	2.015625
bus_routes_7000	6829	15334	4.671875
bus_routes_10000	9767	22758	18.28125
bus_routes_14000	13535	32270	26.609375

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

En primer lugar, la relación entre vértices y arcos de los cuales los arcos pueden ir conectados a los vértices en todas las maneras posibles lo cual implica que es posible tener más arcos que vértices o viceversa, una vez aclarado esto podemos hablar del tiempo de ejecución el cual parte desde un vértice base y busca los caminos más cortos entre 2 vértices con el algoritmo Dijkstra.

El tiempo en ejecución puede justificarse con la complejidad del algoritmo. El algoritmo anterior es recursivo y nos ofrece una complejidad de $O(V^2)$ es decir en número de vértices afecta el tiempo, sin embargo, si queremos relacionar los 3, con la cola de prioridad mínima la complejidad sería de $O(V + A \log V)$ en el cual el número de vértices más la multiplicación de los arcos por el logaritmo de los vértices.

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

Tras ver la implementación del código en el model se puede ver que el grafo siempre es dirigido, es decir, los arcos son unidireccionales. Asimismo, se puede decir que el grafo está fuertemente conectado pues siempre hay un camino entre dos vértices. Este hecho sumado a los resultados de la opción 4 y la ecuación que determina la densidad: $a/v(v-1)$ siendo 'a' numero de arcos y 'v' numero de vértices, es posible decir que el grafo es disperso.

- f. ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo creado en ["connections"] es de 14000 que equivale al número de líneas que tiene el archivo .csv

- g. ¿Cuál es la Estructura de datos utilizada?

Para crear el analizer se usan dos índices de los cuales ['stops'] contiene un MapaADT tipo PROBING y ['connections'] contiene un Grafo dirigido

- h. ¿Cuál es la función de comparación utilizada?

La función de comparación que utilizan ambas estructuras es **compareStopIds** que compara el numero de la parada.