

OBSERVACIONES DEL LA PRACTICA 9

David Alejandro Fuquen 1 Cod 202021113

Juan Andrés Eslava 2 Cod 202012035

Preguntas de análisis

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

Para cambiar el límite de recursión de Python, se usa la función `sys.setrecursionlimit(2 ** 20)`, que en el caso de este lab se encuentra en la línea 158, dentro de el `"if __name__ == '__main__':"`

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

Este cambio es necesario dada la naturaleza recursiva de los algoritmos de búsqueda dentro de los grafos, en particular el DFS. Pues este, es un algoritmo recursivo que llega a recorrer todos los nodos dentro del grafo de manera recursiva, y como en este caso tenemos una cantidad masiva de datos, se requiere de muchas iteraciones.

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

Tal como explica Laboratorio de Matemáticas de la Universidad Autónoma de Madrid (2010/2011)¹, el límite de recursión de Python suele estar inicializado en 1000

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

Creemos que la relación entre vértices, arcos y tiempo es directamente relacionada. Pues, si hay más vértices, habrá más arcos, y el tiempo de ejecución será mayor. Esto se evidencia en la ejecución de la operación con diferentes archivos de cada vez mayor tamaño. Conforme la cantidad de datos incrementaba, así lo hacían los vértices, arcos y tiempo de ejecución.

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

El grafo es denso, pues al crearlo, el tamaño del indegree será equivalente al tamaño original del grafo. El grafo es dirigido (al crearlo, el parámetro `directed` tiene valor `True`). Los elementos del grafo están fuertemente conectados, se ve en la función `connectedComponents()`, en donde se implementa el algoritmo Kosaraju con la función `scc.KosarajuSCC()`.

f) ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo es 14000, se puede ver en el parámetro `"size=14000"`, en la creación del grafo.

g) ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada es `ADJ_LIST`, es decir, lista de adyacencias. Esto se ve al crear el grafo dentro de la función `NewAnalyzer`.

h) ¿Cuál es la función de comparación utilizada?

La función de comparación utilizada fue `compareStopIds`, que compara dos estaciones a partir de su ID.

Referencias:

1. Laboratorio de Matemáticas de la Universidad Autónoma de Madrid(2010/2011), Un Poco de Programación Funcional. Universidad de Madrid. Consultado el 17 de Noviembre de 2021 de: [Un poco de programación funcional — Laboratorio de Matemáticas 2010/2011 \(uam.es\)](https://www.uam.es/~lma20102011/)