

Laboratorio 10

Daniel Reales (201822265) Jose Vicente Vargas (201815601)

Pregunta 1

Enunciado: ¿Que instrucción se usa para cambiar el límite de recursión de Python?

Para determinar el límite de recursión en Python se utiliza la función `setrecursion-limit()`. Esta función es provista por el módulo `sys`. Acorde a la documentación oficial de Python, localizada en la pagina web `docs.python.org`, esta toma un argumento. Este es el nuevo límite de recursión.

Pregunta 2

Enunciado: ¿Por qué considera que se debe hacer este cambio?

Es necesario ajustar el límite de recursión debido a que los algoritmos de recorridos de los grafos tales como *DFS* se basan en llamados recursivos sobre los vertices adyacentes. Debido a que algoritmos basados en DFS (ej: *kosaraju*) necesitan este tipo de comportamiento, es importante tener esta variable en cuenta. De no cambiarla se puede incurrir en errores como un *overflow* del stack de funciones (al llamar recursivamente por encima del límite de memoria determinado) o en un *segmentation fault*.

Pregunta 3

Enunciado: ¿Cuál es el valor inicial que tiene Python como límite de recursión?

El límite por default para las recursiones es de 10^3 . El límite maximo al cual se puede determinar sin ocasionar ningún error es de 10^6 .

Pregunta 4

Enunciado: Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4?

La opción 4 realiza el algoritmo de Dijkstra utilizando como vertice de partida el ingresado por parámetro. Es decir, encuentra la ruta mínima entre la estación base y todos los demás vertices del grafo utilizando este algoritmo.

Debido a que la complejidad temporal de este algoritmo es de $O(E \log(V))$ se espera que (manteniendo el número de vertices constantes) si se aumenta el número de arcos entonces el tiempo crezca linealmente. De igual forma, si se mantiene el número de arcos constante y se incrementa el número de vertices, se espera que la opción crezca en el tiempo de forma logarítmica. Si se incrementan ambas variables al tiempo, entonces se espera un crecimiento “similar” al *linearítmico*.

Pregunta 5

Enunciado: ¿Qué características tiene el grafo definido?

Acorde a la información del archivo *bus_routes_14000.csv* vemos que el grafo tiene las siguientes características:

1. Esta compuesto por 13535 vertices
2. Tiene 32270 arcos
3. Tiene 30 componentes conectados
4. La estación que sirve a más rutas es la *22009* con un total de 32 rutas servidas

Al momento de inicializarlo tiene las siguientes características:

1. Tiene un tamaño inicial de 14000
2. Es un grafo dirigido
3. Esta implementado mediante una lista de adyacencia

Pregunta 6

Enunciado: ¿Cuál es el tamaño inicial del grafo?

Cuando se inicializa el grafo el tamaño inicial es de 14000 y es un grafo *dirigido*.

Pregunta 7

Enunciado: ¿Cuál es la Estructura de datos utilizada?

Se utiliza una *Lista de Adyacencia* para implementar el TAD Grafo

Pregunta 8

Enunciado: ¿Cuál es la función de comparación utilizada?

La funcion de comparación utilizada compara los *id* de las estaciones cuyo valor se encuentra en la llave *key* de cada elemento. Si las llaves son iguales, la función retorna 0, si la llave 1 es mayor a la llave 2 retorna 1 y de lo contrario retorna -1.