

OBSERVACIONES DEL LA PRACTICA

Frank Yasser Ramírez Marín fy.ramirez@uniandes.edu.co, 202215747

2. José Gabriel Bernal Cárdenas jg.bernal1@uniandes.edu.co, 202213421

3. Juan Camilo Gómez Uribe j.gomezu@uniandes.edu.co, 202220238

Preguntas de análisis

- a) ¿Qué instrucción se usa para cambiar el límite de recursión de Python?
-> al final del view se encuentra la siguiente línea de código
`sys.setrecursionlimit(2 ** 20)`
- b) ¿Por qué considera que se debe hacer este cambio?
-> Cambiar el límite de recursión puede ser necesario en casos donde se sabe que los algoritmos necesitan una profundidad/cantidad de recursión mayor. Esto debido a que se podría "desbordar" y generar errores.
- c) ¿Cuál es el valor inicial que tiene Python como límite de recursión?
-> según internet, el valor que tiene python como límite de recursión es 1000
- d) ¿Qué relación existe entre el tiempo que toma la operación 4 con el número de vértices y arcos del grafo?
-> A mayor número de vértices y arcos que tenga el grafo, mayor es el tiempo de carga. Esto bajo la lógica de que tendría que recorrer más datos para entregar una respuesta concreta.
- e) ¿El grafo definido es denso o disperso?
-> el grafo definido es disperso, su densidad está cerca de 0
- f) ¿Qué adaptación debería hacerse a la fórmula en caso de que el grafo fuera NO dirigido?
-> Se le debería quitar la multiplicación de 2 hacia la m. Cosa que es lo mismo a dividirlo por 2
- g) ¿El grafo es dirigido o no dirigido? Para cada tamaño de los datos incluya el resultado de la densidad en el documento de observaciones
-> es grafo es dirigido.
- h) ¿El grafo es conexo?
-> no es conexo, un grafo conexo es donde todos se conectan con todos. Y acá sale 30. Si fuese conexo debería ser 1.

- i) ¿Cuál es el tamaño inicial del grafo?
-> apenas uno carga los datos se da cuenta que el tamaño inicial es de 32270 arcos y 13535 vértices
- j) ¿Cuál es la Estructura de datos utilizada?
-> la estructura de datos utilizada es una ADJ list
- k) ¿Cuál es la función de comparación utilizada?
-> la función de comparación es compareStopIds()