

# OBSERVACIONES DEL LA PRACTICA

Est1 = Samuel Josué Freire Tarazona, 202111460, [s.freire@uniandes.edu.co](mailto:s.freire@uniandes.edu.co)

Est2 = José David Martínez Oliveros, 202116677, [jd.martinez01@uniandes.edu.co](mailto:jd.martinez01@uniandes.edu.co)

## Preguntas de análisis

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

La instrucciones de python necesaria o que puede ayudar a cambiar el limite de recursión es una que se llama `sys.setrecursionlimit`. este moddulo permite defnir el numero que queremos poner como limite. Su visualización en python es de la siguiente manera:

```
sys.setrecursionlimit(default_limit*10)
```

Dentro del parentesis se pone el numero que se quiere definir como nuevo limite de recursion. Este modulo, es del gurpo de funciones modulos, que modfican aspectos del sistema. Este grupo de funciones son llamadas: System specific parameters and function. Dentro del cual existen difernets funciones similares set recursion limit.

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

La implantación de este cambio dentro del código resulta necesaria para el mismo funcionamiento de este. Puesto que, si se logra tener una mayor capacidad de procesamiento de información, se puede conocer mejor el alcance el código. Sin olvidar que si se pone un límite que el computador no aguante, se va a generar un error. Por lo tanto, es necesario hacer este cambio para poder procesar más datos. Además, de que es muy útil para la recursividad de los algoritmos, y eficiencia del mismo código.

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

El valor inicial que tiene Python como límite de recursión es el del número 1000. Es decir, tiene como máximo número de llamadas o recursiones de 1000.

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

Pruebas opción 4

Con el archivo de 50 el tiempo fue de: 15.62 ms vertices: 74 y el número de arcos fue de: 73

Con el archivo de 150 el tiempo fue de: 31.25 ms vertices: 146 y el número de arcos fue de: 146

Con el archivo de 300 el tiempo fue de: 93.75ms vertices: 7295 y el número de arcos fue de: 382

Con el archivo de 1000 el tiempo fue de: 625.00 ms vertices: 984 y el número de arcos fue de: 1633

Con el archivo de 2000 el tiempo fue de: 1906.25ms vertices: 1954 y el número de arcos fue de:3560

Con el archivo de 3000 el tiempo fue de: 3906.25 ms vertices: 2922 y el número de arcos fue de5773

Con el archivo de 7000 el tiempo fue de: 9218ms vertices: 6829 y el número de arcos fue de: 15334

Con el archivo de 10000 el tiempo fue de: 27421.88 ms vertices: 9767 y el número de arcos fue de: 22758

Con el archivo de 14000 el tiempo fue de: 51421.88 ms vertices: 13535 y el número de arcos fue de: 32270

Como se puede ver a lo largo de las pruebas realizadas, el tiempo de ejecución tiene una relación de proporcionalidad con el número de vértices y el número de arcos. No se sabemos cuántos creceré, sin embargo, se puede denotar que el tiempo aumenta mientras lo numero de archivos o vértices aumenta. Por lo tanto, tiene una relación de proporcionalidad como se puede ver en las pruebas de tiempos de ejecución.

Pruebas de opcion 6

Con el archivo de 50 el tiempo fue de: 0.0 ms vertices: 74 y el número de arcos fue de: 73

Con el archivo de 150 el tiempo fue de:0.0 ms vertices: 146 y el número de arcos fue de: 146

Con el archivo de 300 el tiempo fue de:0.0ms vertices: 7295 y el número de arcos fue de: 382

Con el archivo de 1000 el tiempo fue de: 0.0ms vertices: 984 y el número de arcos fue de: 1633

Con el archivo de 2000 el tiempo fue de: 0.0vertices: 1954 y el número de arcos fue de:3560

Con el archivo de 3000 el tiempo fue de: 0.0 ms vertices: 2922 y el número de arcos fue de:5773

Con el archivo de 7000 el tiempo fue de: 0.0 ms vertices: 6829 y el número de arcos fue de: 15334

Con el archivo de 10000 el tiempo fue de: 0.13 ms vertices: 9767 y el número de arcos fue de: 22758

Con el archivo de 14000 el tiempo fue de: 0.15ms vertices: 13535 y el número de arcos fue de: 32270

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

En este caso el grafo se puede considerar denso cuando se coge el tamaño de archivos máximo. Puesto que los arcos, o el número de arcos, está muy cerca del número total o el número de size que se marcó en la estructura inicial. Luego, el grafico es dirigido. Esto se puede ver en su definición donde se dice que la propiedad Direct es TRUE. Por lo que se puede decir que el grafico es dirigido. Además, esta propiedad de dirección nos ayuda a decir que el grafo es fuertemente conectado. Puesto que, si un grafo es dirigido, se puede decir que es fuertemente conectado.

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

El tamaño inicial del grafo como se puede ver en su implementación es de 14000. Esta es la medida inicial que va a tener el árbol en su creación.

```
size=14000,
```

g) ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada fue la de lista adyacentes. Como se puede ver en su implementación. Esta ayuda a guardar información relevante. Esta es muy útil si el grafo es poco denso.

```
datastructure='ADJ_LIST'
```

h) ¿Cuál es la función de comparación utilizada?

En esta implementación se utiliza la función de comparación llamada: compareStopIds. Esta función compara dos entradas, las cuales son dos códigos de información. Y los compara verificando si son mayores, menores o iguales.

```
stopcode = keyvaluestop['key']
if (stop == stopcode):
    return 0
elif (stop > stopcode):
    return 1
else:
    return -1
```