Documento Técnico PRE

PROBLEMA E

ELABORADO POR: JUAN CAMILO ROJAS CATRO

PRESENTADO A: SESBASTIAN CAMILO MARTINEZ REYES

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

AYED

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ D. C.

31 DE AGOSTO DE 2020

**1. Especificación:**

* Entrada:

La entrada son líneas consecutivas, cada con dos números que representan en el rango donde se va a aplicar nuestra función principal, la salida termina cuando no hay más líneas

* Salida

La salida es los números del intervalo y el ciclo más largo entre el rango de números dado.

**2. Estrategia:**

* Descripción y justificación**:**

Para la solución primero tendremos que primero crear el intervalo, una vez lo tenemos creamos un contador de ciclos para el algoritmo que nos daban en la descripción del problema, por último, nos quedábamos con el ciclo que más tardo. Para optimizar el tiempo de respuesta que da el problema agregamos una función de memoria, para esto solo vamos guardando en un diccionario la cantidad de ciclos que tarda el numero que tenemos en el índice de nuestro intervalo, con esto para las próximas recursiones el programa no tarda tanto y es más eficiente. Es importante aclarar que el intervalo hay que ordenarlo por mayor y menor sin embargo en la salida hay que devolverlo en el orden de la entrada.

* Estructura de datos:

Para el problema solo se van a utilizar enteros y diccionarios.

* Algoritmo:

El programa tiene tres funciones cada una centrada en algo, la primera se encarga de leer los datos, enviarlos a buscar en la memoria y encontrar el ciclo máximo, la segunda es la que revisa si el dato pedido esta en la memoria o lo manda a calcular a la tercera, esta ultima es el algoritmo 3n + 1 el cual devuelve la cantidad de ciclos.

**3. Casos de prueba:**

* Entrada:

1 10

100 200

201 210

900 1000

10 1

* Salida:

1 10 20

100 200 125

201 210 89

900 1000 174

1 10 20

* Justificación:

Los problemas base son suficientes para mostrar que el programa sirve y el ultimo nos ayuda a verificar que no existan problemas con

**4. Análisis:**

* Temporal:

Debido a sus ciclos anidados tiene complejidad O(2^n).

**5. Documentación:**

* El nombre del código respectivo es “Problem\_A.py”