Documento Técnico PRE

PROBLEMA A

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 es un numero m cualquiera y finaliza cuando no hay más números.

* Salida

La salida es la cantidad de maneras que hay de “dar las vueltas” a la cantidad m con las monedas de 1, 5, 10, 25.

**2. Estrategia:**

* Descripción y justificación**:**

Para resolver el problema crearemos un arreglo que el índice es la cantidad que necesitamos saber la cantidad de vueltas y so valor es el cantidad de formas, primero definiremos que las formas de 0 es 1 para evitar y a partir de ahí si nuestro monto es mayor que la moneda que tenemos, la cantidad de formas es restarle la moneda al monto + la cantidad de formas restantes (es decir tenemos la moneda actual + más las formas que hay de crear las cantidad restante para llegar al monto), en palabras cortas es sumar las formas actuales más las creadas con la nueva moneda, hacemos esto en el arreglo con todas las monedas .

* Estructura de datos:

Para el problema solo se van a utilizar enteros y listas unidimensionales.

* Algoritmo:

La primera función consiste en leer los datos y la segunda en recorrer las monedas y por cada moneda usar la estrategia nombrada es decir sumar las posibilidades con la nueva moneda y las que ya teníamos. Y devuelve el arreglo terminado, a partir de el primer llamado solo ubica la posición en el arreglo del monto al cual queremos mirar las posibilidades.

**3. Casos de prueba:**

* Entrada:

En la carpeta hay un documento de texto que cuenta con 10000 casos desde 20000 hasta 30000.

* Salida:

La salida se encuentra en otro documento dentro de esta misma carpeta.

* Justificación:

Esta cantidad de pruebas y en ese intervalo es para probar que el programa es lo suficientemente rápido para encontrar la solución a estos casos.

**4. Análisis:**

* Temporal:

Debido a que tiene 3 ciclos anidados la complejidad es O(n^3).

**5. Documentación:**

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