Instituto Politécnico Nacional.

Escuela Superior de Cómputo

ESCOM

Profesor: Joel Omar Juárez Gambino

Alumno:

Torres Abonce Luis Miguel

Grupo:3CM3

Materia: Análisis y diseño de algoritmos

Tema: Problema de la mochila(entero)

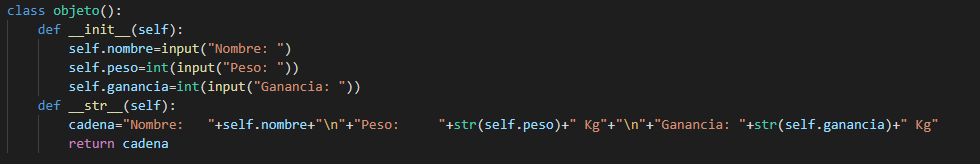
Fecha: 18 de Noviembre 2021

Introducción

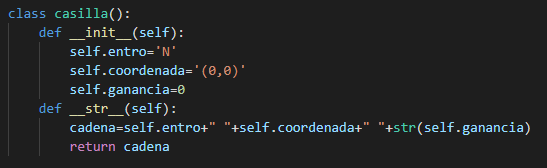
El problema a resolver es el problema de la mochila el cual consiste en que se suponemos tenemos una mochila con cierto peso que puede llevar, además contamos con objetos los cuales tenemos peso y ganancia (que es el valor que se le asigna), la solución lo que hace es verificar todas las opciones posibles para cumplir el objetivo de tener la mayor cantidad de beneficio sin importar el numero de objetos que se metan a la mochila sin sobrepasar su límite de peso.

Desarrollo

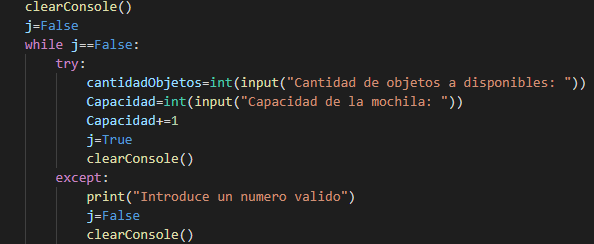
En el programa tengo varias funciones para la resolución del problema así como 2 clases.



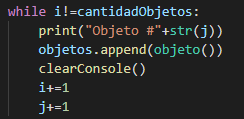
En esta primera clase se tienen a los objetos que se introducirán en la mochila estos cuentan con un nombre, peso y ganancia

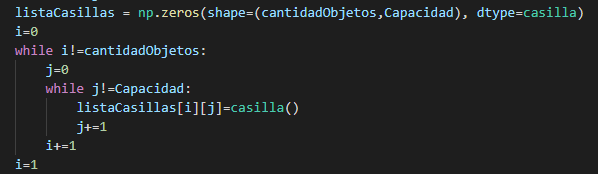


La clase casilla es un objeto que tiene los datos que se necesitan mostrar en la solución en toda la matriz esta cuenta con las variables entro que indica como su nombre lo dice si entra o no, así como las coordenadas de donde proviene la decisión que se toma y la ganancia que se tiene de esa decisión.

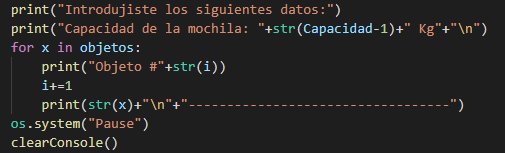


En la función principal es solo el menú que pide los datos que se usaran en la resolución del programa

Este ciclo se utiliza para ir creando los objetos a utilizar.



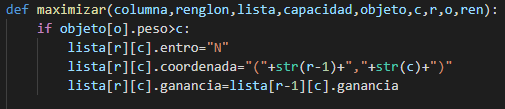
Aquí se crea con la librería numpy la matriz que guardara todos los objetos casilla y con dos ciclos while anidados se crean los objetos en cada uno de los lugares.



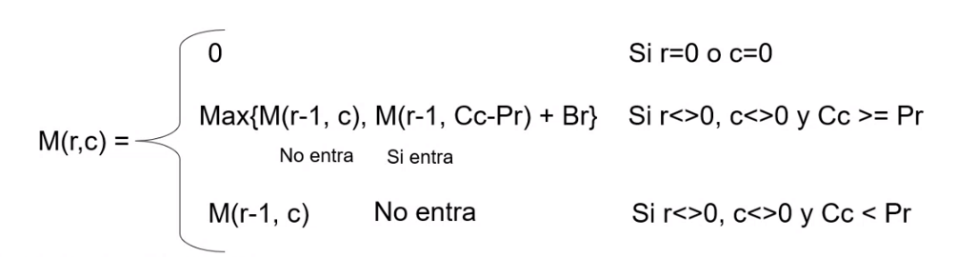
Se imprimen todos los datos que se introdujeron

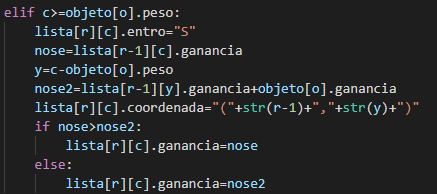


Se manda llamar a la función maximizar, la cual recibe lo parámetro que son:  
columna que es lo mismo que la capacidad, y c que es la capacidad y se envia un 1 para que comience desde esa columna ya que en la columna 0 debe haber un 0, de igual manera con los parámetros renglón y r, además se envia la lista de todas las casilla, de nuevo la capacidad y los objetos y una lista donde se guardaran el numero de objetos que son la respuesta.

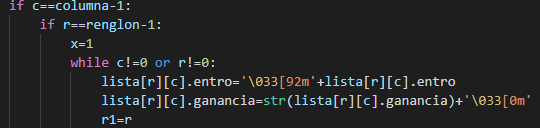


Al principio del programa preguntamos si el objeto del peso es mayor a la capacidad en ese momento de la mochila, si esto se cumple se modifica cada una de sus variables a la correspondiente a la formula

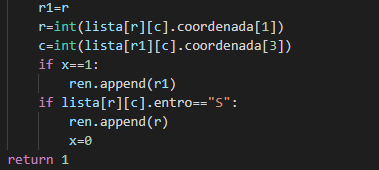




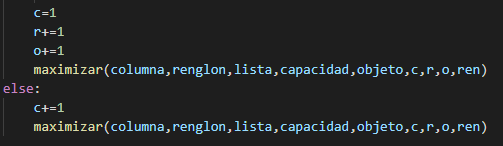
Por el contrario si la capacidad es mayor o igual que el peso del objeto entonces de igual forma se modifican sus variables conforme a la formula, primeramente se obtiene la ganancia del renglón anterior misma columna, también se obtiene la columna restando la capacidad en que es encuentra menos el peso del objeto donde se encuentra y posteriormente se la ganancia de esta casilla en el renglón anterior más la ganancia del objeto en cuestión y se modifica la variable coordenada de la lista en cuestión. Verificamos cual de las 2 ganancias es mayor y se guarda en la variable ganancia.



Verificamos si estamos en la última columna y posteriormente su estamos en la ultimo renglón esto es que estamos en la ultima casilla si esto es verdad, entramos al ciclo while donde si c o r es diferente de r entonces se le cambia el color de letra a toda la casilla para saber que esta es la respuesta



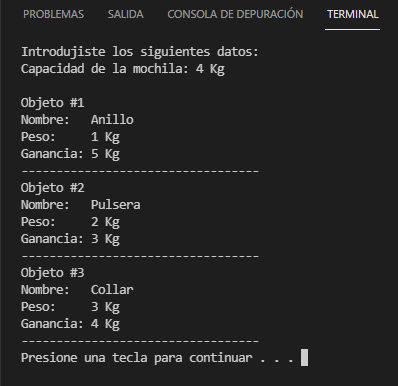
Posteriormente se obtienen las coordenadas donde se tomó la decisión y se igualan a la de renglón y columna de esta forma tenemos la siguiente casilla que se cambiara de color para saber que es la respuesta y se agrega el número de renglón que es la misma variable a objeto a la lista ren y se regresa 1

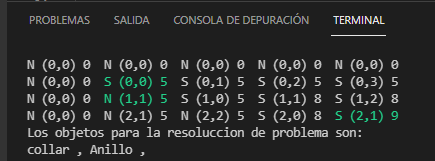


Si estamos en el ultima columna, pero no en el último renglón entonces se reinicia la columna a 1, se le aumenta 1 al renglón y se le aumenta 1 al objeto, y se manda llamar a su misma función, pero con los parámetros agregados.

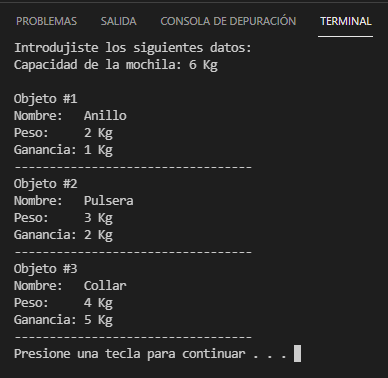
Si no estamos en la última columna entonces solo se le aumenta 1 a la columna y se manda llamar a su misma función con los parámetros agregados.

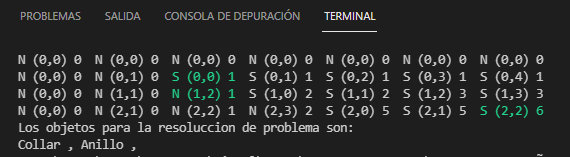
Prueba #1





Prueba #2





Prueba #3

