UNIVERSIDAD AUTONOMA DE NUEVO LEON.

FACULTAD DE INGENIERIA MECÁNICA Y ELECTRICA.

TAREA 5: BIN PACKING

PRESENTADO POR:

JESUS JAVIER MORENO VAZQUEZ 1619830

HORA: V4-V6

PROFESOR: DR. MARIA ANGELICA SALAZAR

SAN NICOLAS DE LOS GARZA, NUEVO LEON, A 27 DE FEBRERO DEL 2016

DESCRIPCION GENERAL DEL PROBLEMA

Se tienen N cantidad de cualquier cosa y se desea empacarse, guardarse o acomodar en la menor cantidad posible de contenedores, en este caso tenemos una lista de 10 canciones, cada una con diferente peso, las cuales se desean empaquetar en discos cuya capacidad máxima es de 10 mb. Lo que se desea es colocar todas las canciones en la menor cantidad de discos posibles,

ALGORITMO PROPUESTO:

1.- Lectura de datos desde CSV.

2.- Ordenar las canciones de manera descendiente respecto al peso.

3.- Tomar la primera canción disponible e introducirla en el primer disco.

4.- Recorrer la lista reordenada de canciones y revisar si no ha sido seleccionado y si su peso no supera al peso disponible en el disco sobre el que estamos trabajando, de ser así se introduce la canción en el disco, de no ser asi, el disco se cierra y se abre otro nuevo

5.- Regresar al paso 3 hasta que todas las canciones estén asignadas a un disco.

6.- Terminar el proceso de asignación y mostrar resultados

bit.py

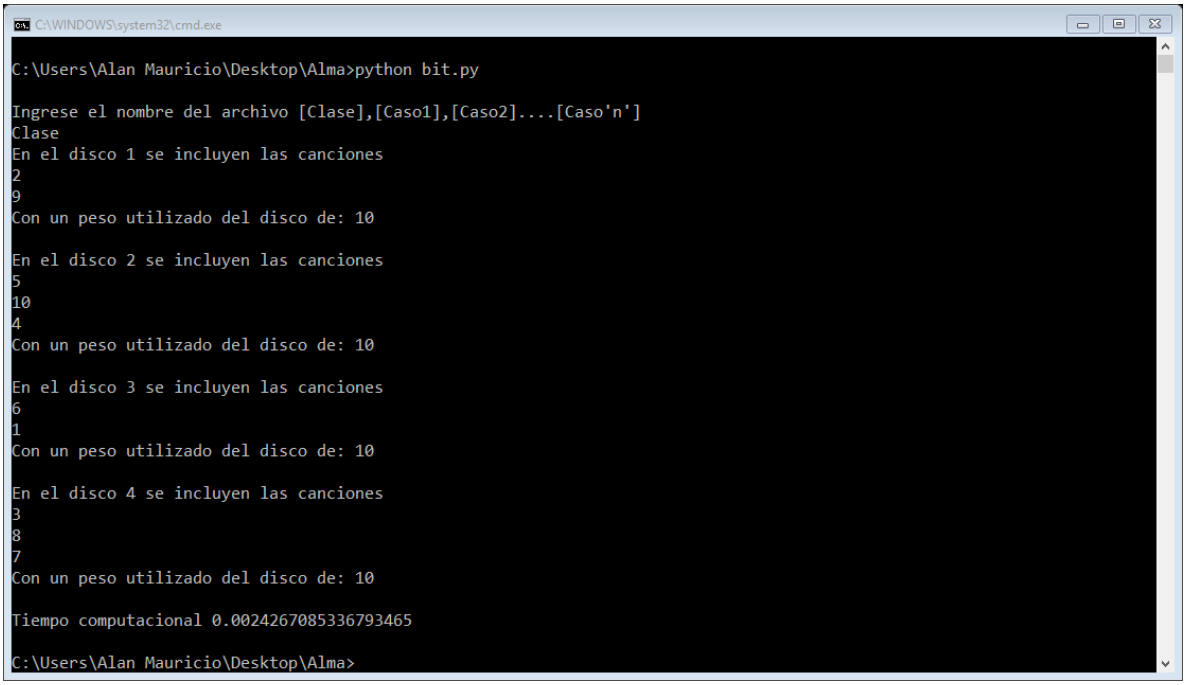
```
            indice_mayor = i
        taken.append(indice_mayor)

    for i in range(len(objetos)):
        valor = objetos[taken[i]]
        lista_reaordenada.append(valor)

    contador_discos = 0
    taken = []
    while(len(taken) < len(lista_reaordenada)):
        auxiliar = []
        i=0
        peso_usado = 0
        peso_disponible = capacidad
        for w in range(len(lista_reaordenada)):
            if(lista_reaordenada[w][1] <= peso_disponible)and not(w in taken):
                valor = lista_reaordenada[w]
                auxiliar.append(valor)
                peso_usado += auxiliar[i][1]
                peso_disponible -= lista_reaordenada[w][1]
                taken.append(w)
                i += 1
        contador_discos += 1
        print("En el disco " + str(contador_discos) + " se incluyen las canciones ")
        for i in range(len(auxiliar)):
            print(str(auxiliar[i][0]))
        print("Con un peso utilizado del disco de: " + str(peso_usado) + "\n    ")
```

RESULTADOS

Para a instancia utilizada, los resultados fueron los siguientes.



- En el disco 1 se incluyen las canciones 2 y 9
- En el disco 2 se incluyen las canciones 5,10 y 4
- En el disco 3 se incluyen las canciones 6 y 1
- En el disco 4 se incluyen las canciones 3,8 y 7

Para esta instancia se aprovechó al máximo la capacidad de los discos utilizados.

CARACTERISTICAS DEL SISTEMA

Procesador: Intel® Core i5-4590 CPU @ 3.30 GHz

Memoria Instalada (RAM): 8.00 GB (7.88 GB utilizable)

Sistema Operativo: Windows 10 Pro

Tipo de sistema: Sistema operativo de 64 bits, procesador x64

CONCLUSIONES

Codificar este algoritmo fue relativamente sencillo, sin embargo, si tuve algunas complicaciones de lógica, por ejemplo, mi asignación se ciclaba y nunca terminaba, tenía un error de índices, así que tuve que crear un índice independiente al ciclo para llevar el control de la condición de mi while. Fuera de eso no tuve ningún problema con el desarrollo de este código.

BIBLIOGRAFIA

- https://www.python.org/doc/