UNIVERSIDAD AUTONOMA DE NUEVO LEON.

FACULTAD DE INGENIERIA MECÁNICA Y ELECTRICA.

TAREA 4: ALGORITMO HEURISTICO PARA PROBLEMA DE MOCHILA

PRESENTADO POR:

JESUS JAVIER MORENO VAZQUEZ 1619830

HORA: V4-V6

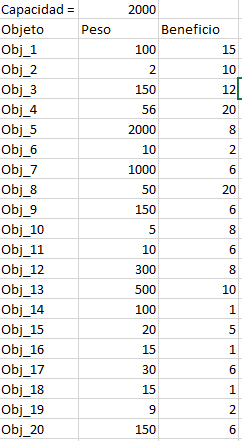
PROFESOR: DR. MARIA ANGELICA SALAZAR

SAN NICOLAS DE LOS GARZA, NUEVO LEON, A 5 DE FEBRERO DEL 2016

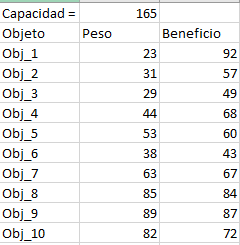
**DESCRIPCION GENERAL DEL PROBLEMA**

Se tiene una lista de ‘n’ objetos, cada uno tiene un peso y un beneficio respectivamente, también se tiene una “mochila” la cual tiene una capacidad ‘K\_max’. El objetivo del problema es introducir tantos objetos a la mochila buscando aumentar al máximo el beneficio sin exceder la capacidad. Se presentan 9 casos para testear el algoritmo desarrollado:

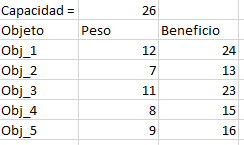
**CASO CLASE**: Propuesto en Clase



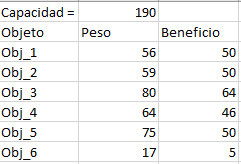
**CASO 1:** 10 Objetos con una capacidad máxima de 165



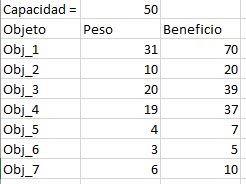
**CASO 2:** 5 objetos y una capacidad 26



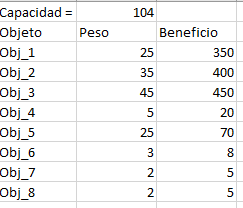
**CASO 3:** 6 objetos y una capacidad de 190

****

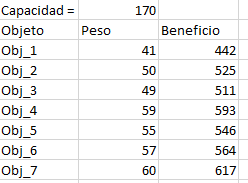
**CASO 4:** 7 objetos y una capacidad de 50



**CASO 5:** 8 objetos y una capacidad de 104



**CASO 6:** 7 Objetos con una capacidad de 170

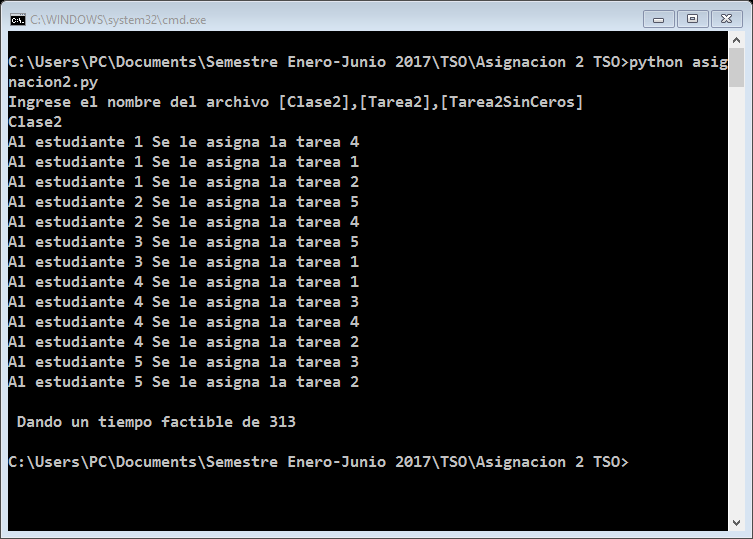


**ALGORITMO PROPUESTO**

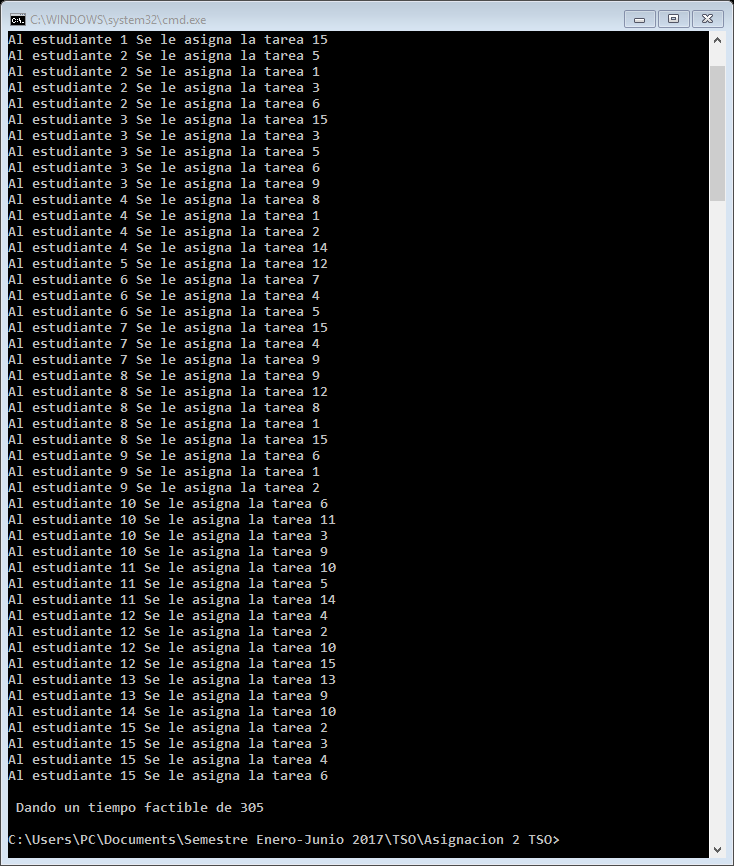
1. Lectura de datos
2. Selecciona los valores mínimos de cada estudiante hasta cumplir con el mínimo de tareas que debe realizar, sin tomar en cuenta las actividades que no pueda realizar en caso de que así sea.
3. Si aún no se cumplen con las demandas de cada tarea, volverá a recorrer los datos, pero ahora por las tareas que aún no se cumplan
4. Seleccionará el menor verificando que no rebase el máximo de tareas que puede realizar el estudiante, en caso de que no quede disponible ese estudiante, pasara al segundo menor y así hasta encontrar uno disponible.
5. Se suman los tiempos de las tareas seleccionadas
6. Se reporta el resultado obtenido.

**RESULTADOS**

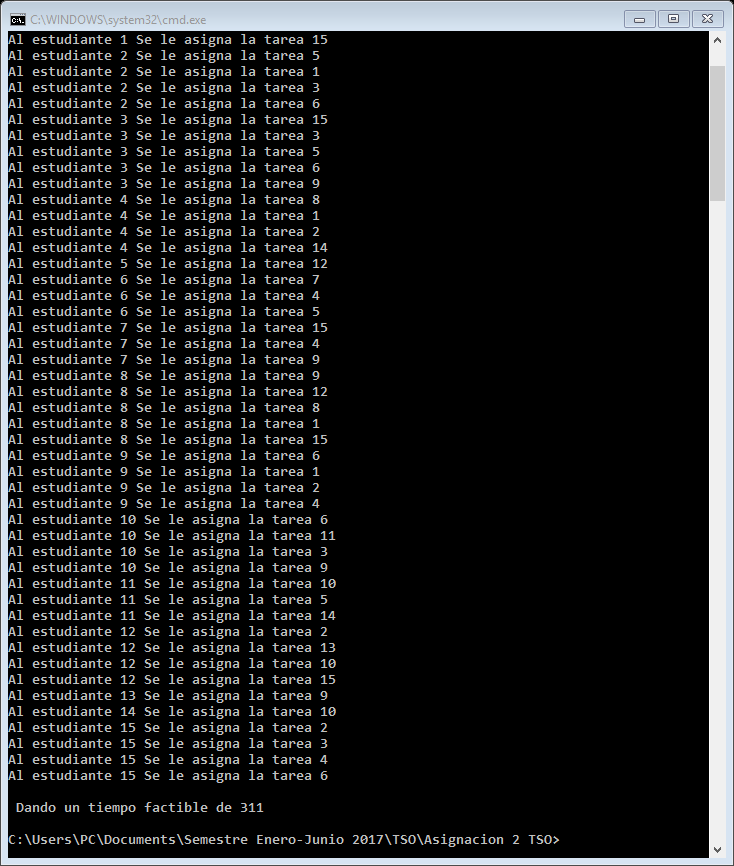
Caso 1:



Caso 2: Proporcionado por la docente sin restricciones en individuos



Caso 3: Proporcionado por la docente con restricciones en individuos.



**CONCLUSIONES**

Codificar este algoritmo fue un reto, el manejo de variables y controladores fue más difícil que en la primera tarea, cuando logré por fin terminarlo me dio un buen tiempo con respecto al promedio, sin embargo, al realizar las pruebas de escritorio me di cuenta que no estaba tomando en cuenta el número de tareas máximas que puede realizar un individuo en específico, algunos individuos excedían el número de tareas que podían realizar, cuando corregí eso el tiempo aumentó. Recomiendo realizar pruebas de escritorio con ejercicios similares para corroborar que el algoritmo funciona como se supone que debe funcionar.

**BIBLIOGRAFIA**

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