UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

Ingeniería en Computación

DISEÑO Y ANALISIS DE ALGORITMOS

Tarea 4 Knapsack

Profesor: Marcelo Pérez Medel

Leonardo Olvera Martínez

Grupo: 1507

Fecha: martes 24 de octubre de 2023

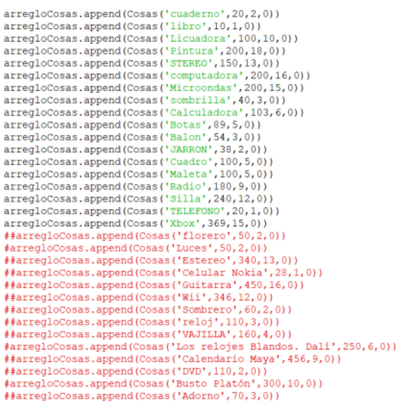
Tarea No. 4 Programa que resuelve el problema del Knapsack

1era Parte

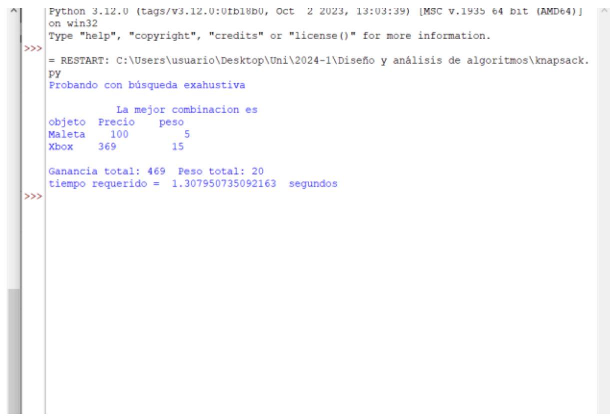
En la plataforma se encuentra ya el programa que resuelve el problema del saco del ladrón o Knapsack, con una serie de artículos, pruebe lo siguiente:

1.- Para qué valor de n su computadora se tarda cerca de 1 segundo

Mi computadora aproximadamente en el valor de n = 18 se tarda 1.3079 segundos



```
arregloCosas.append(Cosas('cuaderno',20,2,0))
arregloCosas.append(Cosas('libro',10,1,0))
arregloCosas.append(Cosas('Licuadora',100,10,0))
arregloCosas.append(Cosas('Pintura',200,18,0))
arregloCosas.append(Cosas('STEREO',150,13,0))
arregloCosas.append(Cosas('computadora',200,16,0))
arregloCosas.append(Cosas('Microondas',200,15,0))
arregloCosas.append(Cosas('sombrilla',40,3,0))
arregloCosas.append(Cosas('Calculadora',103,6,0))
arregloCosas.append(Cosas('Botas',89,5,0))
arregloCosas.append(Cosas('Balon',54,3,0))
arregloCosas.append(Cosas('JARRON',38,2,0))
arregloCosas.append(Cosas('Cuadro',100,5,0))
arregloCosas.append(Cosas('Maleta',100,5,0))
arregloCosas.append(Cosas('Radio',180,9,0))
arregloCosas.append(Cosas('Silla',240,12,0))
arregloCosas.append(Cosas('TELEFONO',20,1,0))
arregloCosas.append(Cosas('Xbox',369,15,0))
##arregloCosas.append(Cosas('florero',50,2,0))
#arregloCosas.append(Cosas('Luces',50,2,0))
##arregloCosas.append(Cosas('Estereo',340,13,0))
##arregloCosas.append(Cosas('Celular Nokia',28,1,0))
##arregloCosas.append(Cosas('Guitarra',450,16,0))
##arregloCosas.append(Cosas('Wii',346,12,0))
##arregloCosas.append(Cosas('Sombrero',60,2,0))
##arregloCosas.append(Cosas('reloj',110,3,0))
##arregloCosas.append(Cosas('VAJILLA',160,4,0))
#arregloCosas.append(Cosas('Los relojes Blandos. Dalí',250,6,0))
##arregloCosas.append(Cosas('Calendario Maya',456,9,0))
##arregloCosas.append(Cosas('DVD',110,2,0))
#arregloCosas.append(Cosas('Busto Platón',300,10,0))
##arregloCosas.append(Cosas('Adorno',70,3,0))
```



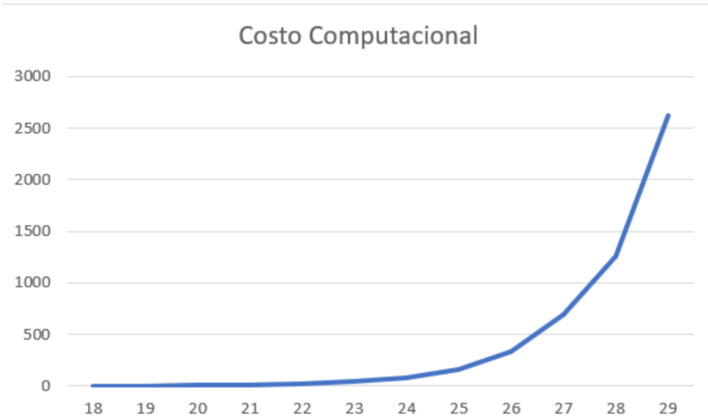
```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\usuario\Desktop\Uni\2024-1\Diseño y análisis de algoritmos\knapsack.
py
Probando con búsqueda exahustiva

           La mejor combinacion es
objeto  Precio    peso
Maleta    100        5
Xbox    369        15

Ganancia total: 469  Peso total: 20
tiempo requerido =  1.307950735092163  segundos
>>>
```

2.- Descomente a partir de este punto 10 objetos más de uno por uno y haga una tabla y una gráfica de estos datos, ¿cómo crece la gráfica?

18	19	20	21	22	23	24	25	26	27	28	29
1.3079	2.3292	4.4365	8.8632	18.8281	39.52447	78.1699	163.8087	329.8393	693.5308	1252.603	2624.8483



La grafica crece de manera exponencial, de manera más específica con una formula:

$$O(n) = 2^n$$

3.- Calcule cuantos objetos puede procesar en no más de una hora

29 objetos, ya que se tarda, aproximadamente, 2624 segundos, es decir, 43 minutos y 44 segundos, teniendo en cuenta que el algoritmo duplica el tiempo anterior, con 30 objetos tardaría aproximadamente 1 hora y 26 minutos, se pasa.

4.- ¿Cuánto tiempo tardaría su computadora en procesar toda la lista?

Para 5 objetos tardaría 20998 segundos, es decir 5 horas, 49 minutos y 58 segundos.

5.- ¿Cuánto tiempo se tardaría en procesar 50 elementos?, ¿y 100 elementos?

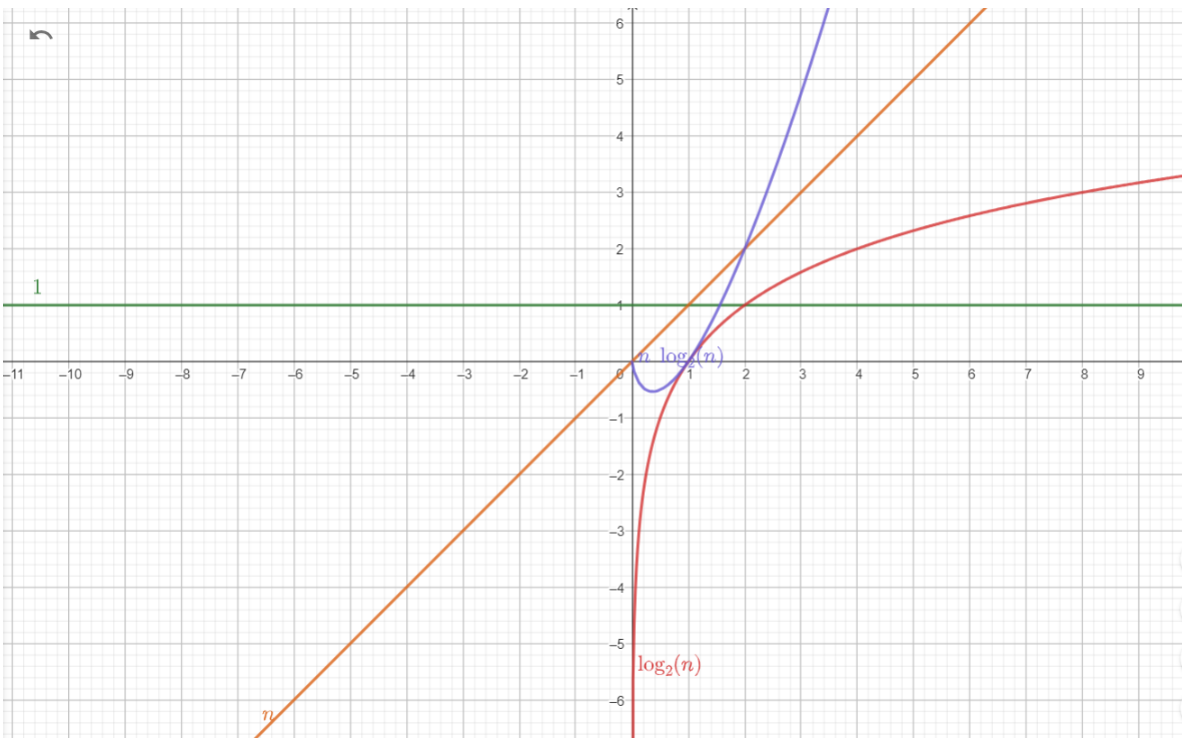
Para 50 elementos tardaría aproximadamente 5,504,705,862.04 segundos, es decir 174 años, 201 días y poco más.

Para 100 elementos tardaría aproximadamente 6,197,747,817,268,680,000,000,000.00 segundo, es decir 1,965,292,940,534,208.5 siglos… Mucho tiempo.

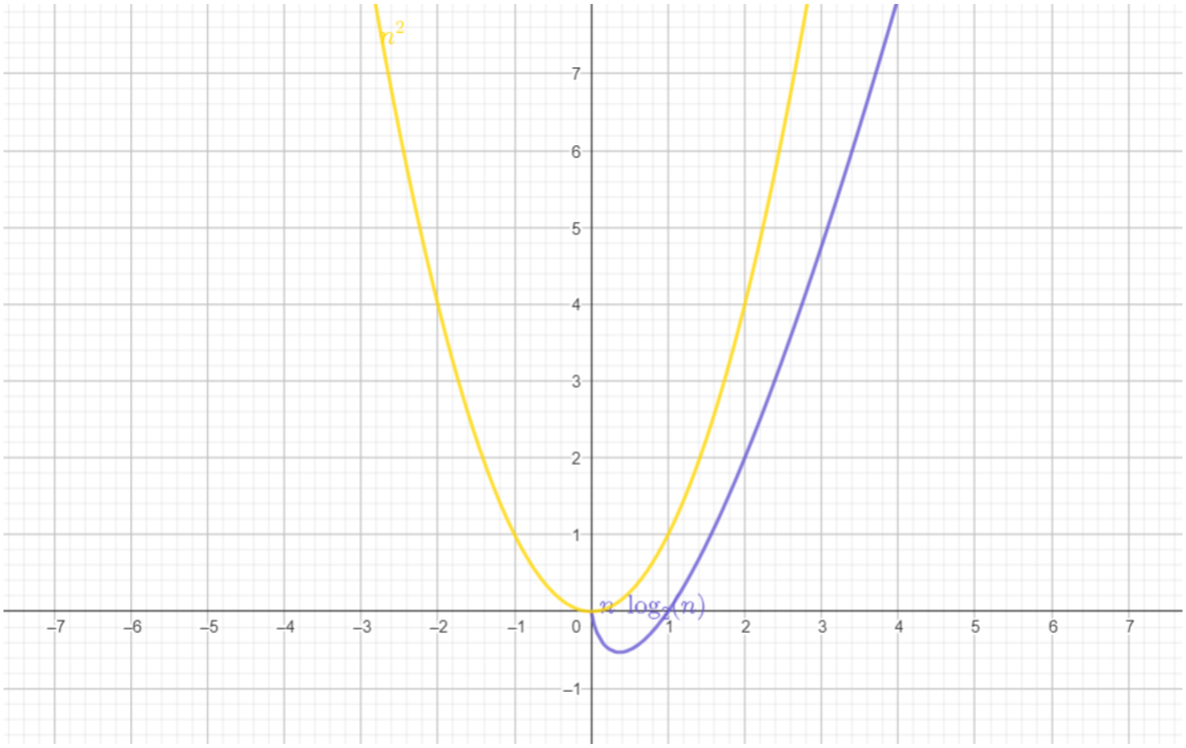
2da parte

Realice las siguientes gráficas en GeoGebra o el programa de su elección:

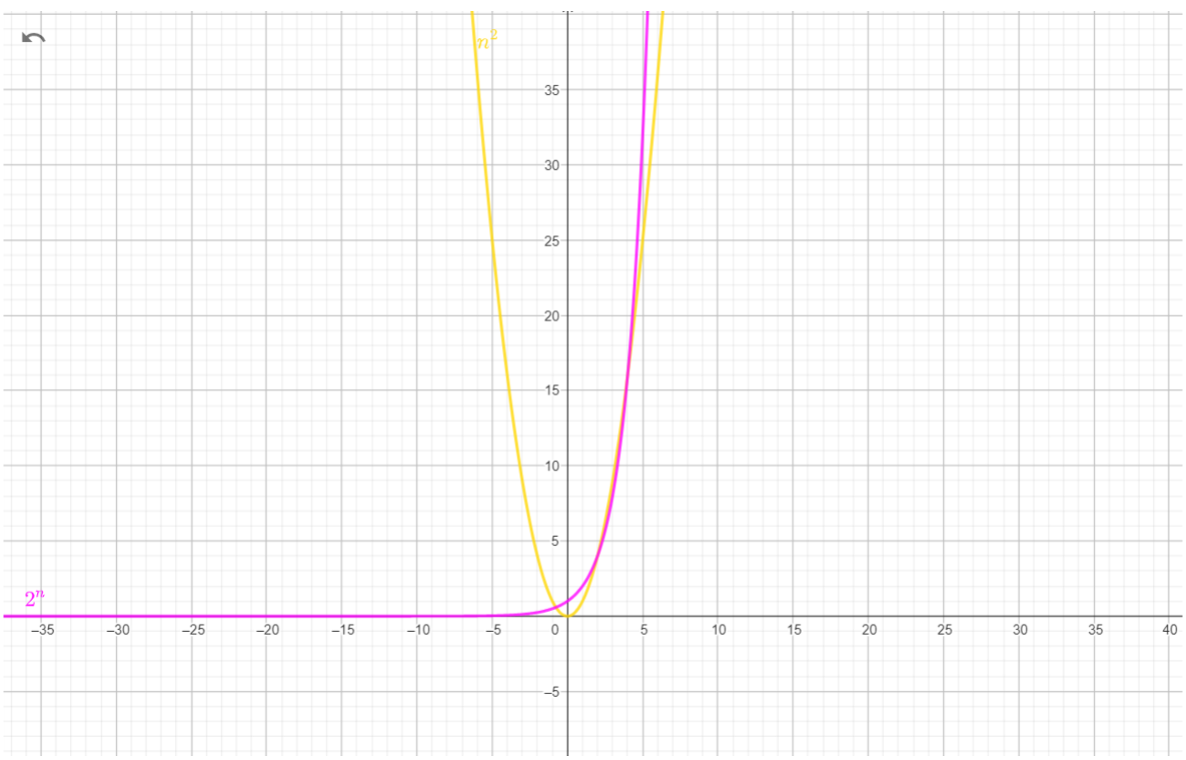
a.- Grafique O(1), O(log n), O(n) y O(n log n) todas en la misma gráfica



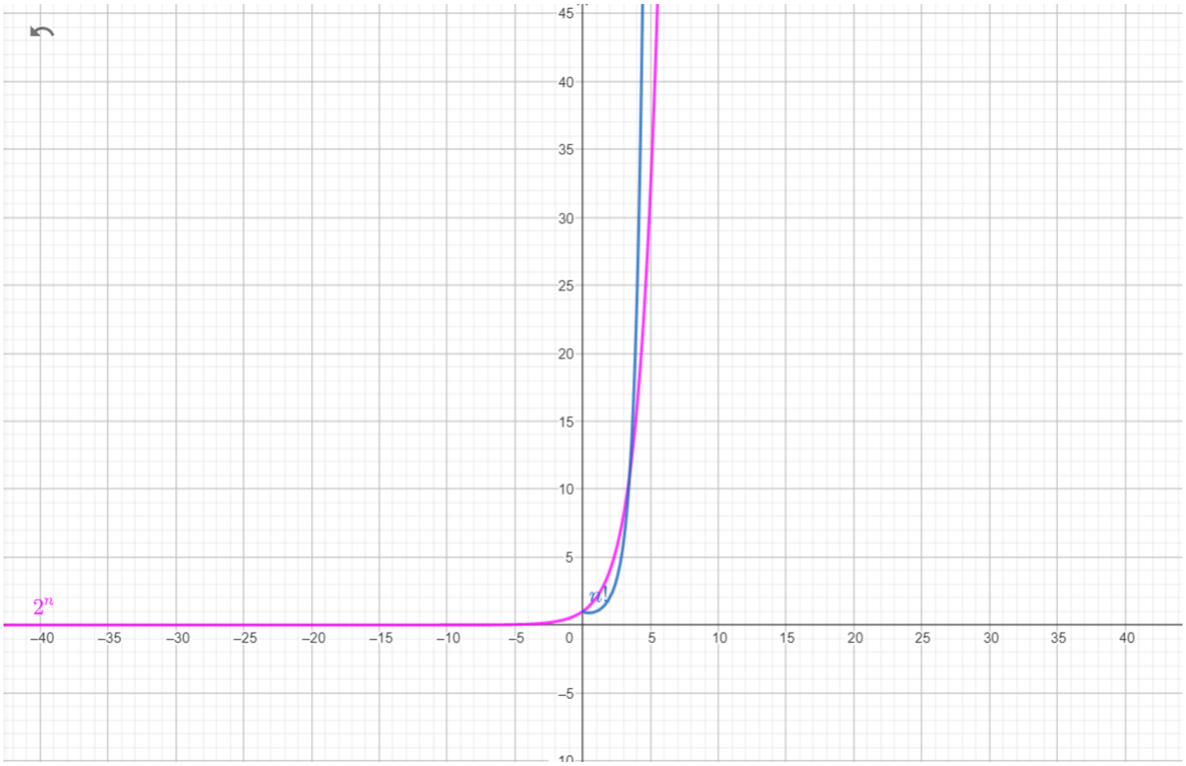
b.- Grafique O(n log n), O(n^2)



c.- Grafique O(n^2) y O(2^n)



d.- Grafique O(2^n) y O(n!)



3era Parte

Vea los siguientes dos videos. Has un pequeño resumen de media cuartilla para cada uno, de lo que personalmente consideres más importante:

https://www.youtube.com/watch?v=1x4VbYerGsA&t=128s - El Mayor Problema de la Computación SIN RESOLVER

Hacía ya 100 años desde que el matemático David Hilbert propuso 23 problemas matemáticos cuando el instituto Clay Mathematics Institute of Cambridge propuso 7 problemas nuevos, los cuales hasta le fecha aún están en resolución porque aun hay cosas que las matemáticas aun no pueden resolver o aun no se encuentra la forma.

El problema del que se habla en el video es el problema P vs NP, específicamente se entiende que los algoritmos computacionales requieren una determinada cantidad de tiempo y de recursos para poder ejecutarse, el problema se centra en el tiempo, y es que depende de como se realice el algoritmo puede costar mas o menos tiempo. La letra P se refiere a “polinómico” y hace referencia a aquellos problemas o algoritmos que pueden resolverse en un tiempo polinómico, es decir, el tipo de orden debe ser de monomios. NP, por el contrario, significa no polinómico, es decir, que su orden es de tipo exponencial (mucho más costoso) pero si queremos comprobar que sea correcto, esta comprobación es de tipo polinómica.

Teniendo en cuenta lo anterior es que se plantea la pregunta, ¿Es posible que P = NP? si los problemas de orden no polinómico se pueden comprobar de manera polinómica, ¿podría haber alguna manera de resolverlos de manera polinómica?

https://www.youtube.com/watch?v=2Xkv-W9tOXU&t=4s - La destacable historia detrás del algoritmo más importante de todos los tiempos

El video habla sobre la importancia de la Transformada Rápida de Fourier (FFT) en varios campos, especialmente en la detección de pruebas de armas nucleares subterráneas. La Transformada Rápida de Fourier (FFT) es un algoritmo fundamental que se utiliza en una variedad de aplicaciones, desde radares y sonares hasta tecnologías 5G y Wi-Fi. Fue descubierto por científicos que intentaban detectar pruebas de armas nucleares encubiertas, y su adopción temprana podría haber frenado la carrera armamentista nuclear.

Después de que Estados Unidos lanzara bombas atómicas en Hiroshima y Nagasaki, propuso el "Plan Baruch", que buscaba un control internacional de los materiales radiactivos para uso pacífico, pero los soviéticos lo rechazaron. Esto dio inicio a la carrera armamentista nuclear global. Las pruebas nucleares se llevaron a cabo en lugares remotos, como las islas del Ártico y el sur del Pacífico, y tuvieron efectos negativos, incluyendo la radiación de los propios estadounidenses.

Se pasó de las armas de fisión a las bombas termonucleares, que eran mucho más poderosas y peligrosas. En 1954, una prueba de hidruro de litio resultó en una explosión mucho más grande de lo esperado, causando la irradiación de personas cercanas. Se realizaron conferencias y negociaciones para prohibir las pruebas nucleares, pero el mayor desafío fue verificar si ambas partes cumplirían sus acuerdos.

La Transformada Rápida de Fourier (FFT) redujo significativamente la cantidad de cálculos necesarios para analizar las señales de sismómetros y permitió la detección de pruebas nucleares subterráneas. A pesar del desarrollo de la FFT, la prohibición amplia de pruebas nucleares no se logró a tiempo, y muchas naciones continuaron realizando pruebas subterráneas.

El video también menciona que el matemático Carl Friedrich Gauss había descubierto una forma de resolver la transformada discreta de Fourier antes que otros, pero su descubrimiento se pasó por alto debido a la falta de publicación adecuada y la comprensión limitada en ese momento.