	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. MARCO ANTONIO MARTINEZ

Asignatura: ESTRUCTURA DE DATOS Y ALGORITMOS

Grupo: 17

No de Práctica(s): 11

Integrante(s): ISLAS ESPINO JESÚS ABRAHAM

No. de Equipo de cómputo empleado: 23

No. de Lista o Brigada:

Semestre: 2020-2

Fecha de entrega: 2020

Observaciones:

CALIFICACIÓN: _____

Práctica 11
Estrategias para la construcción de algoritmos

Objetivo

El objetivo de esta guía es implementar, al menos, dos enfoques de diseño (estrategias) de
algoritmos y analizar las implicaciones de cada uno de ellos.
construir algoritmos
(fuerza bruta, algoritmo ávido, bottom-up, top-down, divide y vencerás, etc).

INTRODUCCIÓN

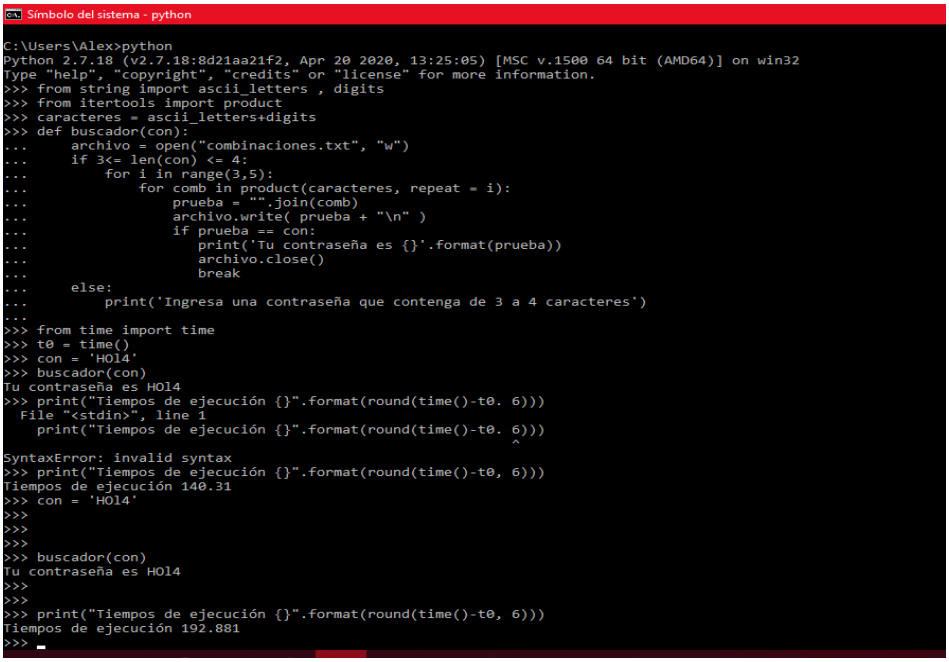
A través de los años, los científicos de la computación han identificado diversas técnicas generales que a menudo producen algoritmos eficientes para la resolución de muchas clases de problemas.
En la actualidad el ser humano se enfrenta a distintas situaciones: de aprendizaje, de retroalimentación y en muchas ocasiones dificultades que con la experiencia o por la elección de la alternativa apropiada, va dando solución.

Esta práctica se presentaran algunas de las técnicas más importantes como son: (fuerza bruta, algoritmo ávido, bottom-up, top-down, divide y vencerás,

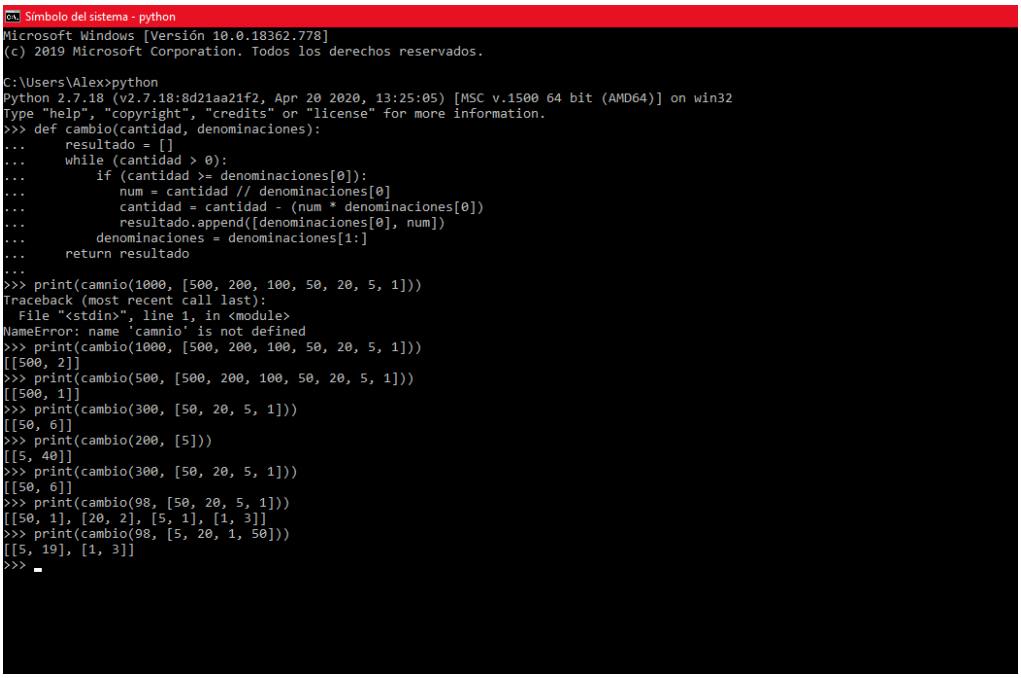
Para resolver un problema se pueden desarrollar diversos algoritmos, existen en ocasiones múltiples soluciones, pero dentro de ellas existen las que son más eficientes y es aquí donde la habilidad del desarrollador juega un papel importante. Por lo que la práctica continua contribuye a la mejora del desarrollo.

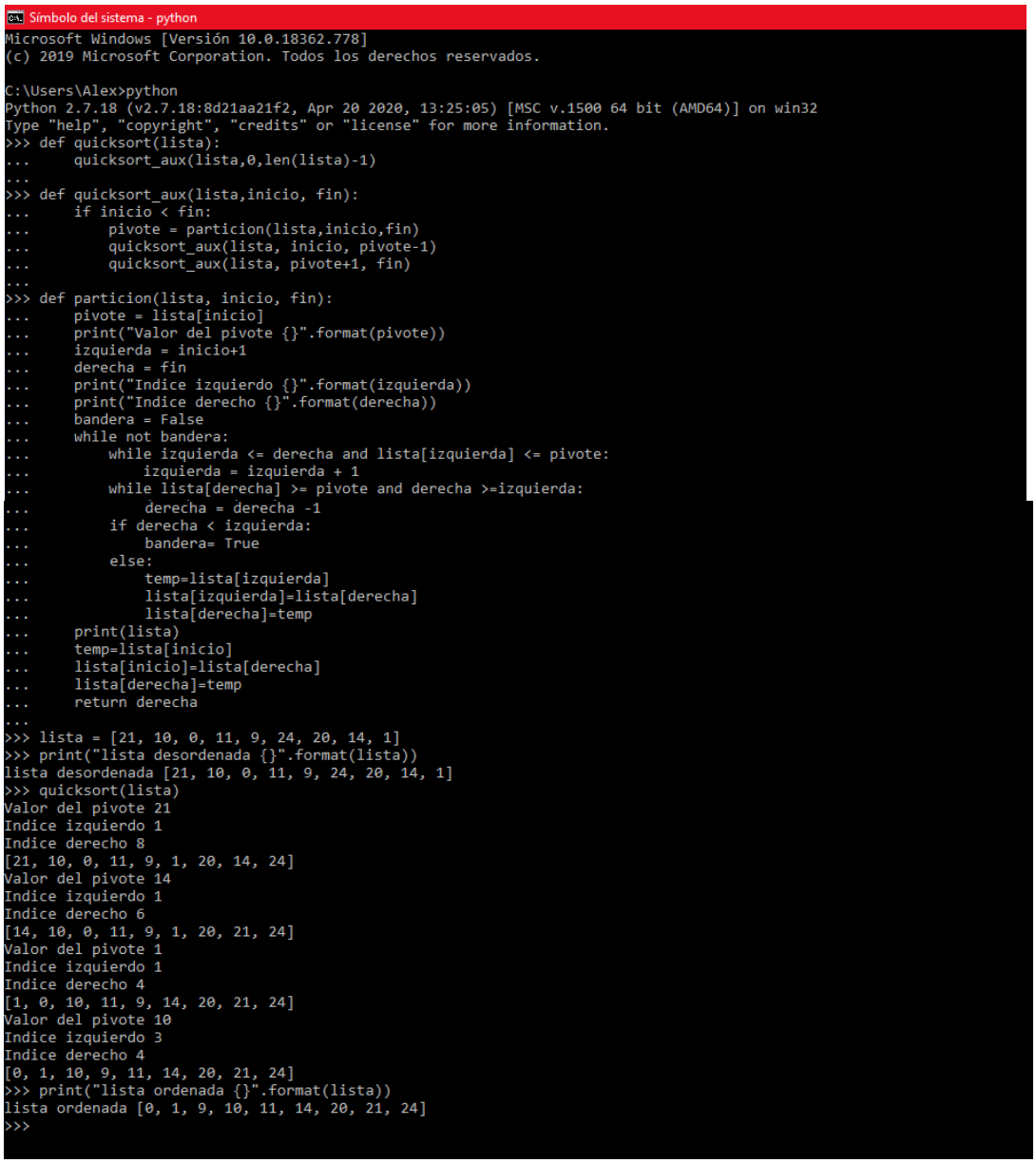
DESARROLLO

FUERZA BRUTA



AVIDO





Explicación de código "Divide y Vencerás":

Comenzando por decir que este está compuesto por 3 funciones donde una de ellas se llama así mismas cuando el programa se está ejecutando, los elementos que usa son el pivote que es el primer elemento del arreglo.

Para lograr hacer el ordenamiento este ejecuta la función quicksort(lista) haciendo que de esta manera se le pase a arreglo en el cual va a trabajar, hablando de la la función quicksort_aux esta comparar que el inicio se mayor al fin con la finalidad de realizar el arreglo en el intervalo dado, siguiendo por la variable pivote que se iguala a la función partición siendo esta función la que contiene el algoritmo de divide y vencerás, lo último que se realiza es que la función

quicksort_aux se llama de nuevo así misma pero ahora en intervalos de 2 grupos ya formados.

Conclusión

La práctica me sirvió para familiarizarme con estos algoritmos y revisar el comportamiento que tienen y el uso que se le puede dar a cada uno dependiendo la aplicación que se necesitó.