|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Marco Antonio Martinez Quintana |
| *Asignatura:* | Estructura de Datos y Algoritmos I |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 11 |
| *Integrante(s):* | Abrego Abascal Diego |
| *No. de Equipo de cómputo empleado:* | - |
| *No. de Lista o Brigada:* | 1 |
| *Semestre:* | 2 |
| *Fecha de entrega:* | 28/04/2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Estrategias para la construcción de algoritmos**

Introducción

Al encontrar problemas y desafíos que enfrentar siempre se tienen muchas más probabilidades de conseguir resultados efectivos si las problemáticas son analizadas desde un análisis con la profundidad necesaria. De dicho análisis se puede elegir un plan de acción u otro según sea el mas conveniente para el escenario en cuestión. Para elegir el plan de acción mas adecuado es de vital importancia conocer las estrategias desde las cuales se puede abordar el problema. Estrategias las cuales serán revisadas durante la elaboración de esta práctica.

Fuerza Bruta

Esta estrategia consiste principalmente en encontrar la respuesta al problema proporcionado realizando una búsqueda exhaustiva de todas las combinaciones y variantes posibles. Esta estrategia resulta ser efectiva si se dispone del poder de computo necesario y si el problema a resolver de un tamaño razonable, debido a que fácilmente se pueden presentar una cantidad enorme posibles combinaciones. La clara desventaja de esta estrategia es que el tiempo de ejecución puede llevar un tiempo razonable.

Algoritmos Ávidos

Estos se caracterizan por seguir una serie de ordenes en un orden especifico. Sin embargo, dichas ordenes o condiciones solo son llevadas a cabo y evaluadas una sola vez, lo que brinda velocidad al algoritmo a cambio de no siempre brindar la solución más eficaz al problema a resolver.

Bottom-up

Esta consiste en tratar de resolver el problema final resolviendo las partes más pequeñas y básicas del problema y que al unirlas, la suma de las partes pequeñas brinde la solución total a la problemática.

Top-down

Esta, en comparación con la estrategia anterior, empieza a calcular la solución desde la posición más alta del problema hacia abajo. Junto con la estrategia Top-down se suele usar la técnica de memorización, la cual consiste en ir guardando los resultados de los cálculos ya realizados para posteriormente evitar tener que realizar operaciones y regresar un resultado de manera mucho más eficaz.

Incremental

Consiste en incrementar paulatinamente una o mas variables, comprobar que estas sean correctas y seguir incrementando y comprobando de manera paulatina para llegar a la solución completa del problema.

Divide y vencerás

Por último, esta estrategia consiste en separar la problemática total en tareas más chicas y sencillas para así afrontar el problema parte por parte y al final llegar a la solución de manera efectiva.

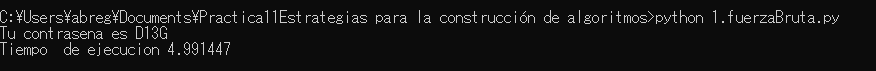
Objetivo

“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.”

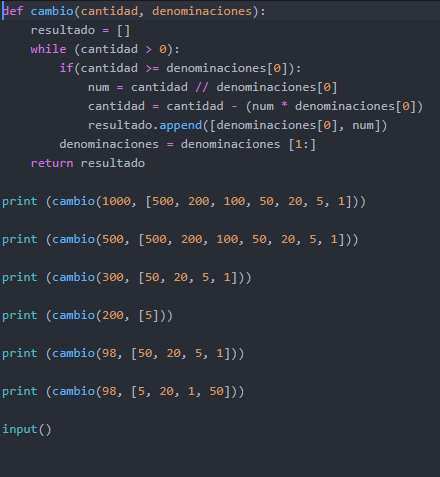
Desarrollo

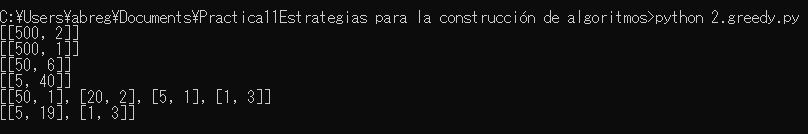
1. Fuerza Bruta



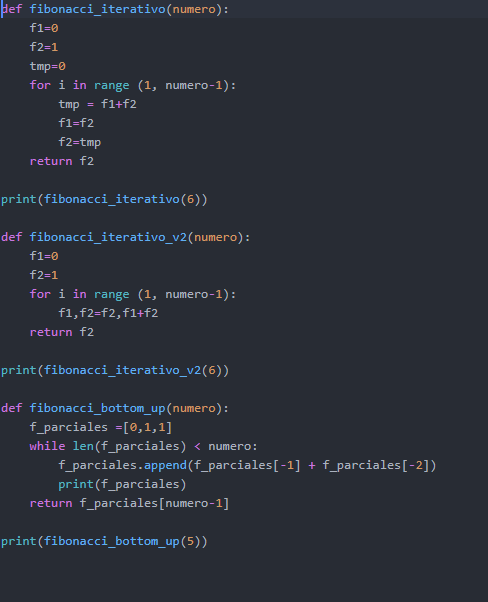


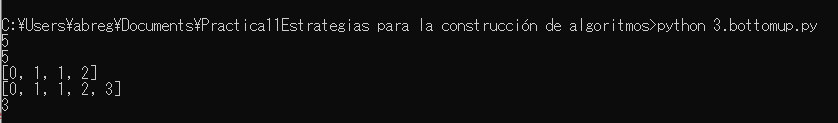
1. Algoritmos Avidos





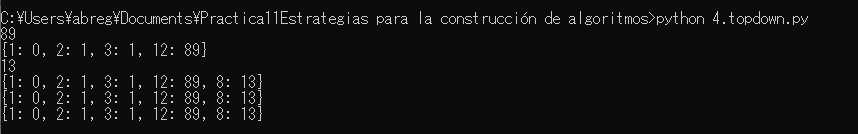
1. Bottom-up



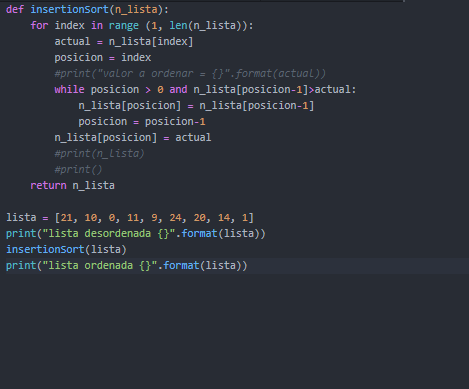


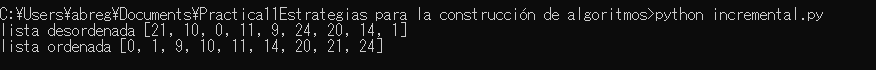
1. Top-down



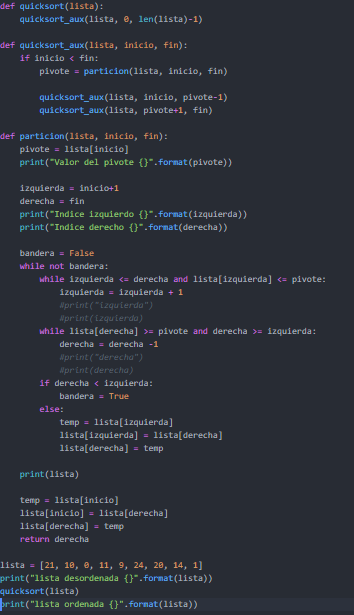


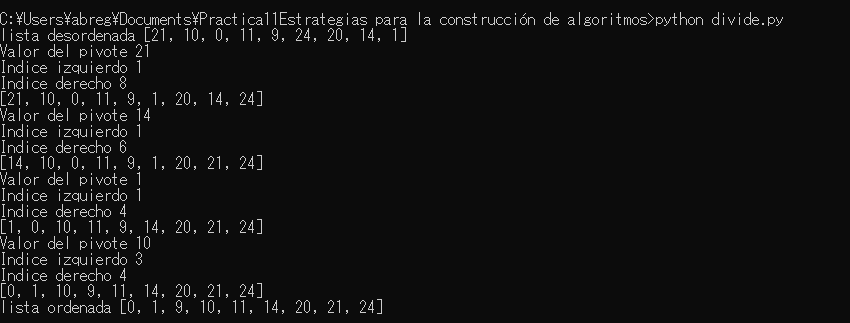
1. Incremental



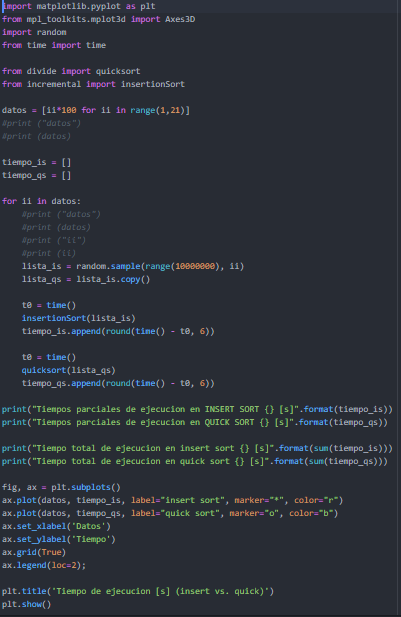


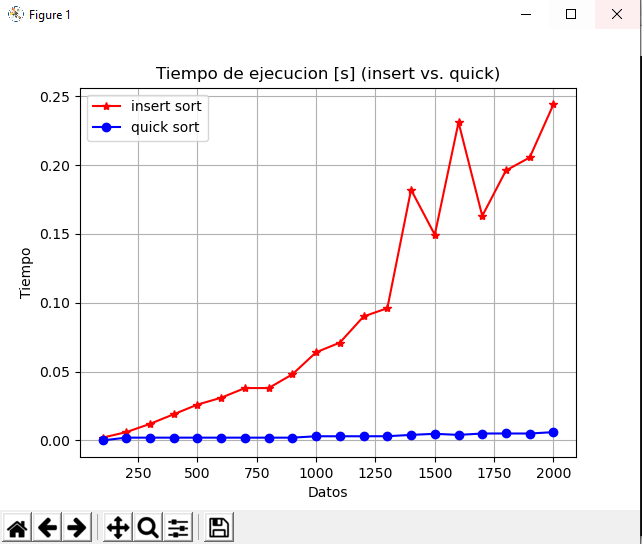
1. Divide y Vencerás

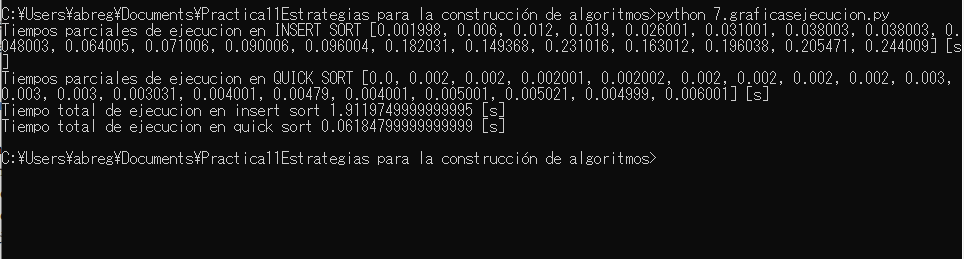




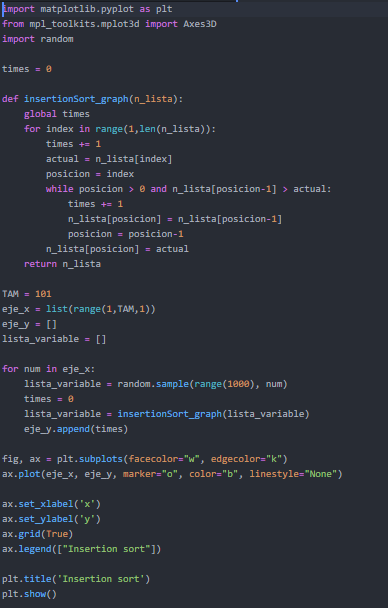
1. Graficando de Tiempos de Ejecución

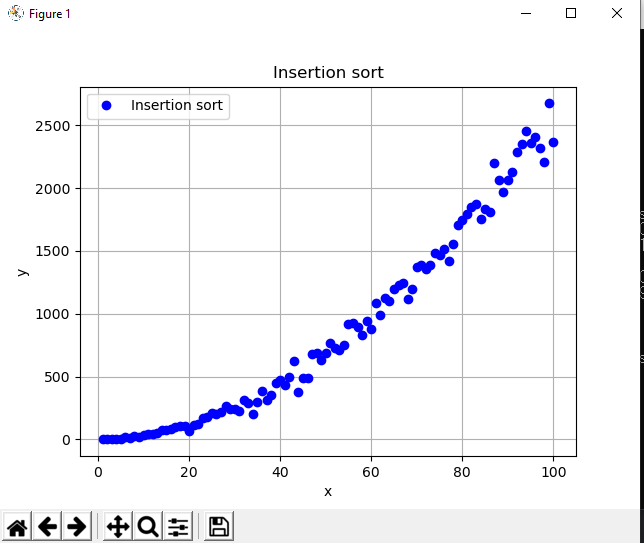






1. Modelo RAM





Conclusión

Con la realización de esta practica se revisaron importantes estrategias sobre las cuales se pueden abordar las problemáticas a resolver. Con dichas estrategias se elaboraron diversos ejercicios con los cuales se comprobó y aprendió en que escenarios una estrategia brilla y bajo qué circunstancias es mejor usar otra, denotando las ventajas y desventajas de cada una.

Entre las estrategias revisadas, la primera fue la de fuerza bruta, para la cual se diseño un algoritmo que mediante todas las combinaciones posibles lograba descifrar una determinada contraseña. Sin embargo, al tratar de que esta descifrara contraseñas de 5 caracteres el tiempo de ejecución del programa aumentaba considerablemente dando a notar la desventaja que esta estrategia presenta cuando se enfrenta a un número elevado de combinaciones que realizar. Para el algoritmo ávido, se hizo un algoritmo que tenia que repartir cambio, sin embargo, al modificar las condiciones para que no diera un resultado tan directo, este ya no arrojaba soluciones optimas, por lo que demostró su ineficiencia según que escenarios. Para los siguientes ejercicios se usaron diversas estrategias para ordenar una lista de números, y posteriormente se graficó el tiempo de ejecución de las diversas estrategias para mostrar cual era la más eficaz al realizar esta tarea, donde el algoritmo que se mostró más eficiente fue el que ordeno la lista con quick sort.

Por lo mencionado anteriormente y por lo obtenido con la realización de esta práctica, concluyo que los objetivos se cumplieron correctamente.

Bibliografía

* http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS\_EDA1/eda1\_p11.pdf
* https://www.python.org