PRÁCTICA 4

Algoritmos de búsqueda. Parte 1.

## Alumno: Robles Martínez Héctor

## Número de Cuenta UNAM: 317356548

## Semestre: Tercer semestre de Ingeniería en Computación (2016)

## Semestre Actual: 2021-1

## Profesor: Jorge A. Solano G.

Objetivo:

Identificar el comportamiento y características de algunos algoritmos de búsqueda por

comparación de llaves.

Instrucciones:

• Implementar búsqueda secuencial en Python tanto de forma iterativa como de forma

recursiva para encontrar un nodo (con el parámetro de búsqueda que desees).

• Implementar búsqueda binaria en Python tanto de forma iterativa como de forma

recursiva para encontrar un nodo (con el parámetro de búsqueda que desees). Antes

de realizar la búsqueda se debe utilizar un método de ordenamiento directo (n 2 ) y

uno logarítmico (n ∗ log(n)).

• Obtener la complejidad algorítmica para cada implementación (búsqueda secuencial,

búsqueda binaria con ordenamiento directo y búsqueda binaria con ordenamiento

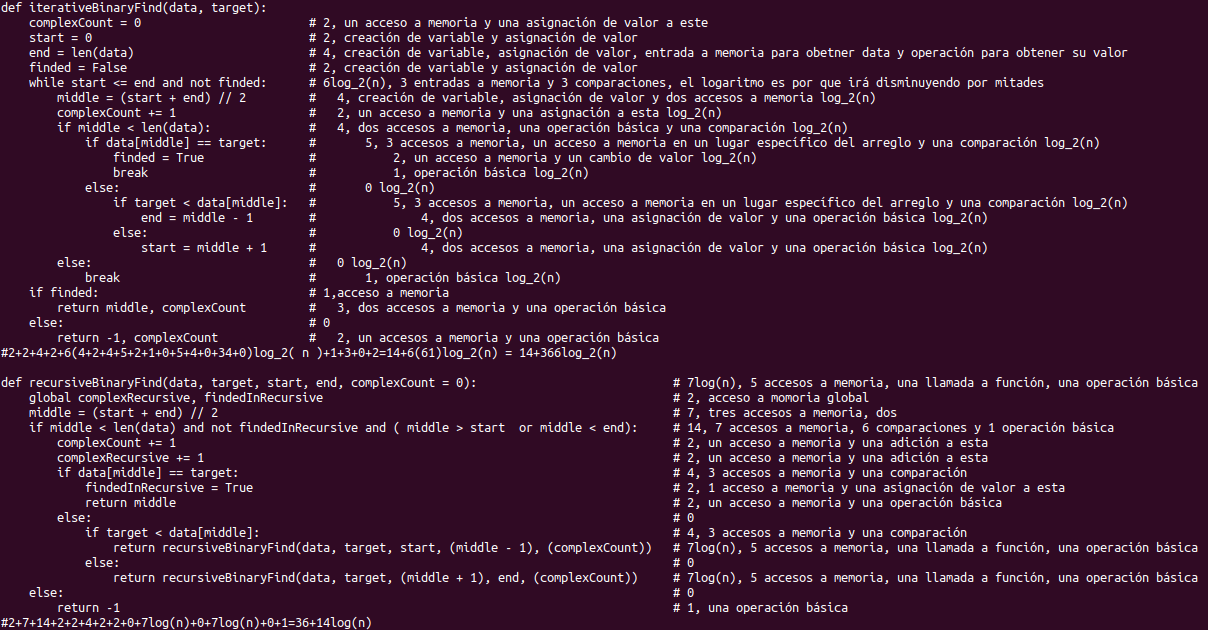
logarítmico).

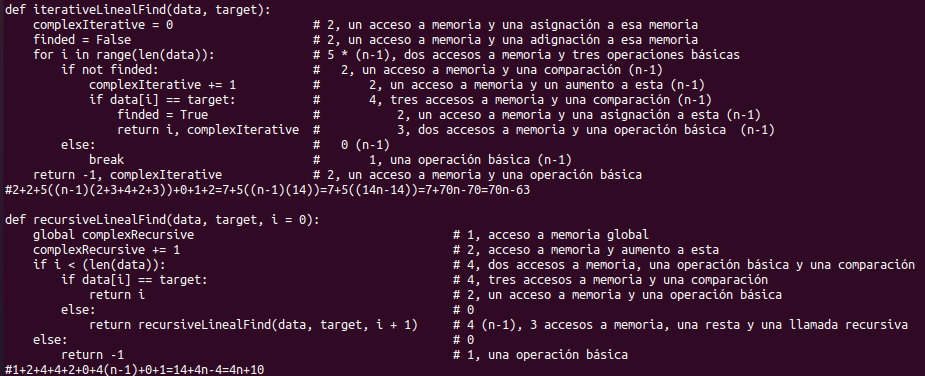
• Graficar el comportamiento de los algoritmos para el mejor, el peor y el caso prome-

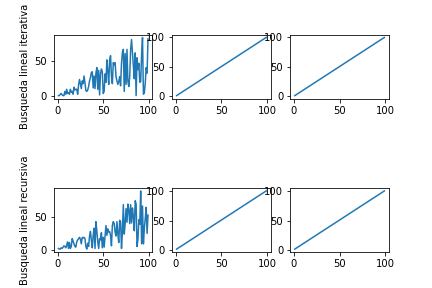
dio para cada implementación (búsqueda secuencial, búsqueda binaria con orde-

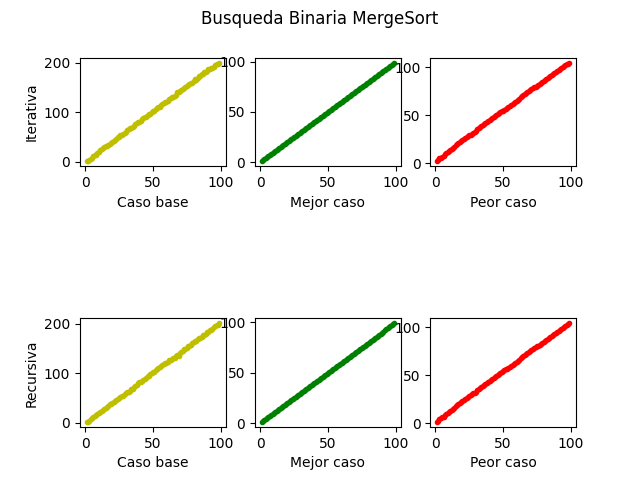
namiento directo y búsqueda binaria con ordenamiento logarítmico).

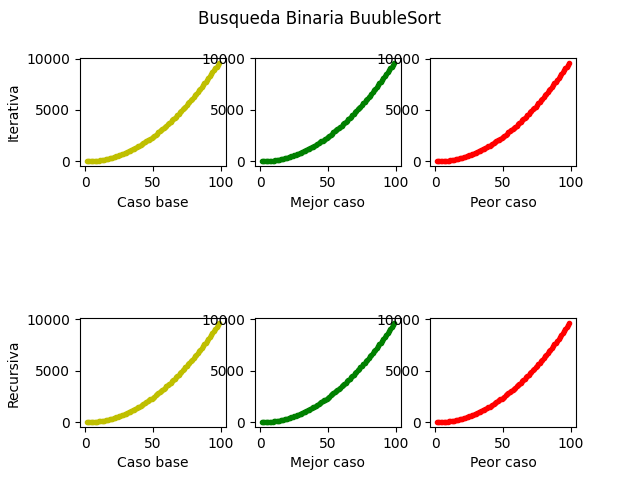
Resultados obtenidos:

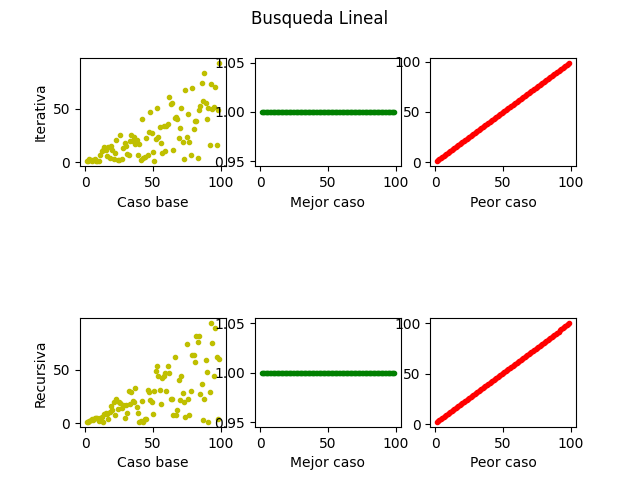












Conclusiones:

A pesar de que el algoritmo de búsqueda binaria tiene un tiempo de búsqueda logarítmica, se debe conciderar el tiempo del algoritmo de ordneamiento, el cual puede llegar a tener cualquier complejidad, quitando por completo su eficiencia a este algoritmo, mientras que el lineal, como su nombre lo dice, es lineal, es fácil de implementar pero puede llegar a ser muy tardado para conjuntos grandes.