PRÁCTICA 3

Algoritmos de ordenamiento. Parte 3.

## 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:

Conocer e identificar la estructura de los algoritmos de ordenamiento Counting Sort y

Radix Sort.

Instrucciones:

• Implementar en Python los algoritmos de Counting Sort y Radix Sort.

• Obtener los polinomios del mejor, el peor y el caso promedio de complejidad de cada

algoritmo (Counting sort y Radix sort).

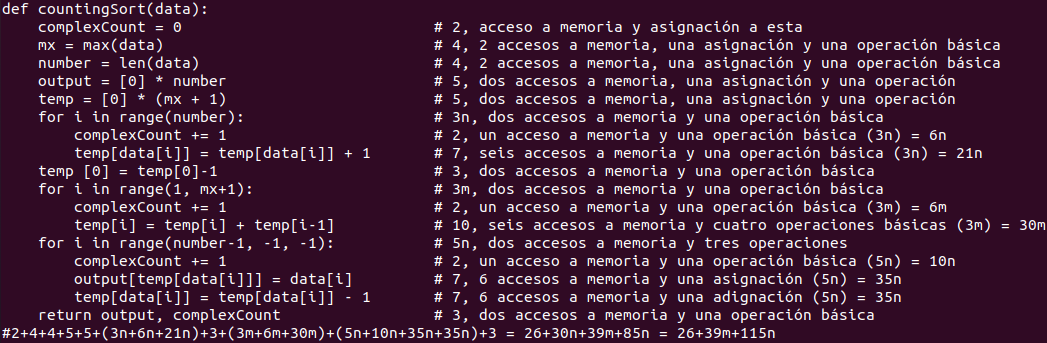
• Graficar en Python el comportamiento de los algoritmos implementados para difer-

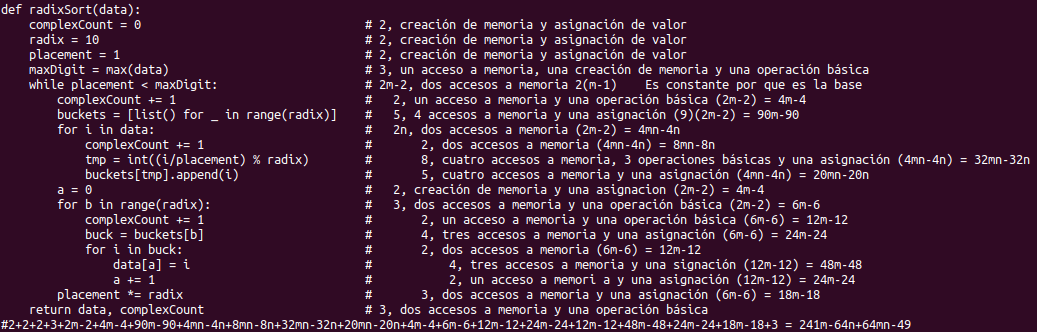
entes instancias de tiempo (listas de 1 a 1000 elementos) para el mejor (lista orde-

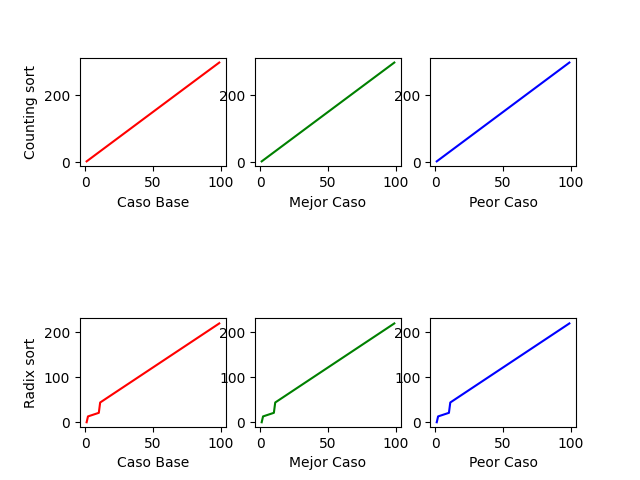
nada), el peor (lista ordenada en forma inversa) y el caso promedio (lista aleatoria)

de complejidad.

Resultados obtenidos:







Conclusiones:

Ambos son algoritmos de ordenamiento rápios y eficientes, su único problema esque sólo pueden ordenarse números en el caso de Counting Sort y Se desperdicia mucha memoria en el caso de Radix Sort