Laboratorio 3

Juan Camilo Monterrosa Sanchez

October 14, 2018

Assume that A is an array of size n of distinct elements

1 Minimum number of inversions - instance

Para un arreglo A de tamaño n, la instancia donde el ordenamiento creciente tiene la cantidad minima de inversiones, es un arreglo ordenado, del tipo:
A = [1 , 2 , ... , n-2 , n-1 , n]

2 Maximum number of inversions - instance

Para un arreglo A de tamaño n, la instancia donde el ordenamiento creciente tiene la cantidad Maxima de inversiones, es un arreglo completamente desordenado, done $n * (n - 1)/2$, del tipo: A = [n , n-1 , ... , 3 , 2 , 1]

3 Complexity (worst case number of comparisons) of the brute force counting on A

Para un arreglo A de tamaño n, mediante el conteo por "Fuerza Bruta", la instancia donde la complejidad es la mayor posible, es en un arreglo completamente desordenado, done O(n^2), del tipo:
A = [n , n-1 , ... , 3 , 2 , 1]

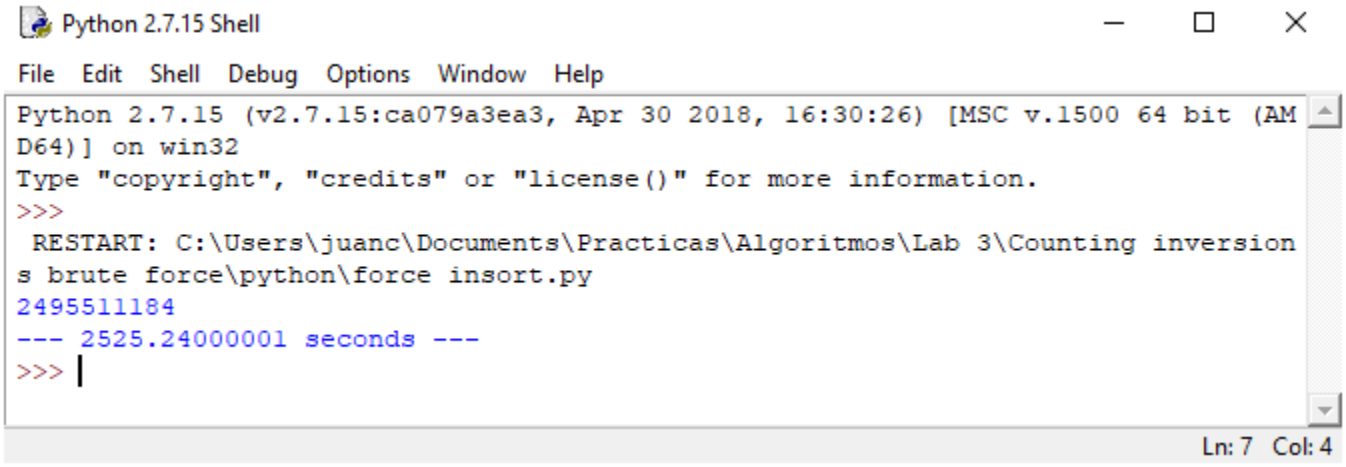
4 Complexity (worst case number of comparisons) of the divide an conquer (mergesort) counting on A

Para un arreglo A de tamaño n, mediante "MergeSort", la instancia donde la complejidad es la mayor posible, es en un arreglo completamente desordenado, done O(n*log(n)), del tipo:
A = [n , n-1 , ... , 3 , 2 , 1]

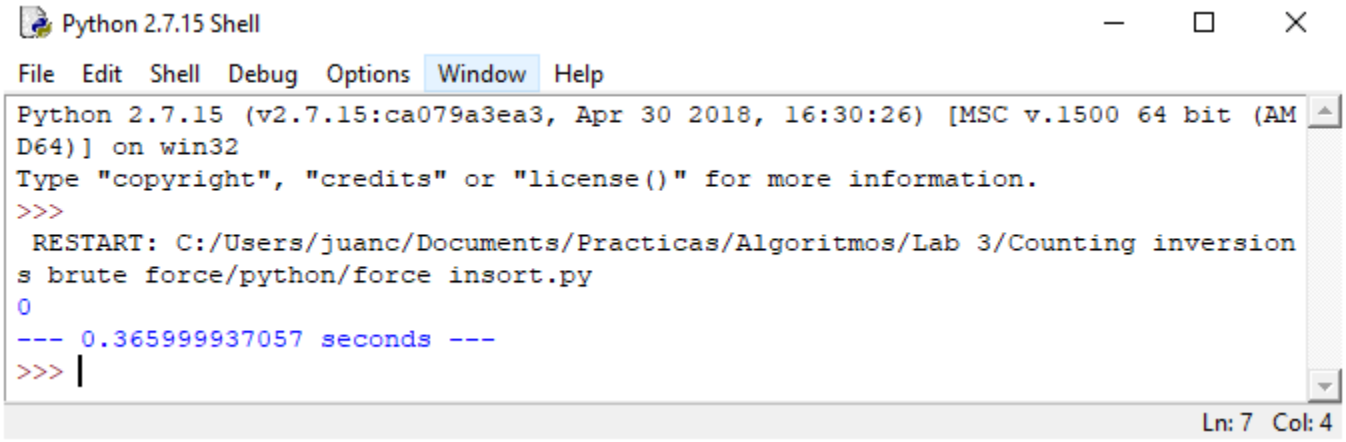
5 Run in your local machine the brute force and divide and conquer algorithms in Python 2.7

calculate the time for the first 10^5 numbers of size instance from:

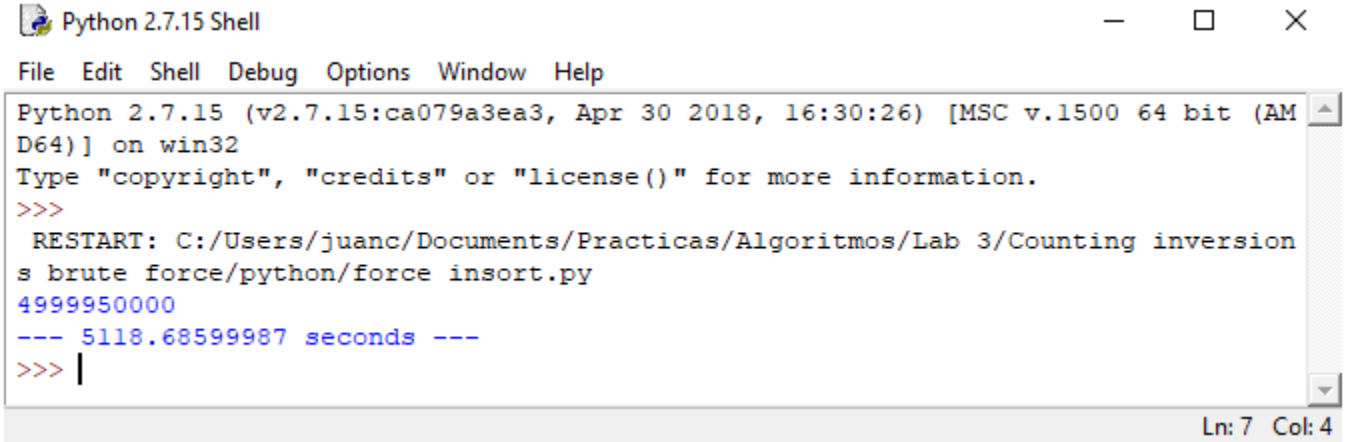
Hackearth:



Sorted increasing:



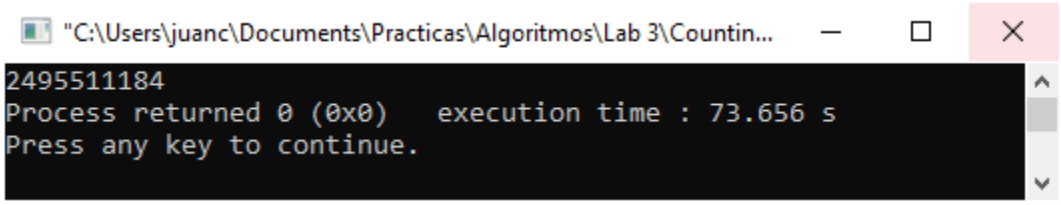
Sorted decreasing:



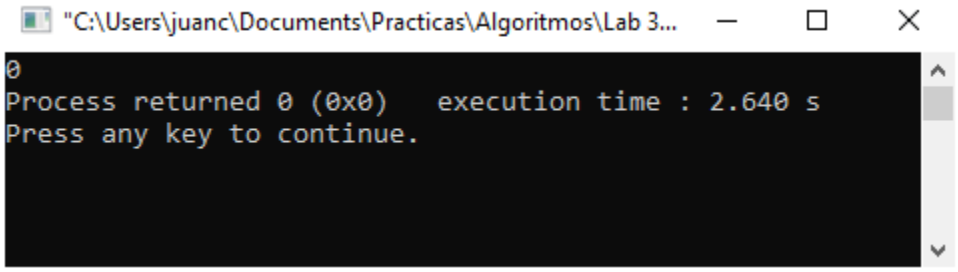
6 Run in your local machine the brute force and divide and conquer algorithms in C or C++

calculate the time for the first 10^5 numbers of size instance from:

Hackearth:



Sorted increasing:



Sorted decreasing: