ITSC - Extensión áulica Villa El Libertador
Programación III Trabajo práctico evaluativo

Temas :

- Notación O
- Análisis de complejidad de algoritmos
- Algoritmos de ordenamiento

Actividades :
1) Implementar la función selection_sort en el archivo ss.py
2) Implementar la función insertion_sort en el archivo is.py
3) Escribir tests para ambas funciones de ordenamiento, considerando (al menos) los siguientes casos:
○ lista vacía
○ lista con un elemento
○ lista ordenada con dos elementos
○ lista desordenada con dos elementos
○ listas con más de dos elementos
○ lista de diferentes tipos (con números, con caracteres, con tuplas)
4) Escribir el análisis de la cantidad de comparaciones que realiza cada algoritmo
5) Determinar la complejidad de cada algoritmo considerando mejor caso y peor caso
6) Realizar mediciones usando el módulo timeit para listas de diferentes tamaños y graficar los valores obtenidos para el mejor caso y el peor caso
7) Evaluar si los resultados obtenidos coinciden con el análisis hecho en los puntos 4 y 5.

Respuestas

4)Ordenamiento por Seleccion
entrada= lista de elementos comparables
salida=una lista ordenada que es permutacion de la lista original

PASO 0:
L=[-4,2,8,7] i=0, j=1, pos_min=0

*comparo L[j] con L[pos_min] → j=2
*comparo L[j] con L[pos_min] → j=3
*comparo L[j] con L[pos_min] → j=4
intercambio L[i] con L[pos_min]

PASO 1:
L=[-4,2,8,7] i=1, j=2, pos_min=1
*comparo L[j] con L[pos_min] → j=3
*comparo L[j] con L[pos_min] → j=4
intercambio L[i] con L[pos_min]

PASO2:
L=[-4,2,8,7] i=2, j=3, pos_min=2
*comparo L[j] con L[pos_min] → j=4
intercambio L[j] con L[pos_min]

Ordenamiento por insercion

Peor de los casos: que este ordenada exactamente al reves
L=[6,4,3,2,1,0]

PASO 1: insertar el 4
* comparar el 4 con el 6 → intercambio
[4,6,3,2,1,0]

PASO 2: insertar el 3
* comparar el 3 con el 6→ intercambio
[4,3,6,2,1,0]
* comparar el 3 con el 4 → intercambio
[3,4,6,2,1,0]

PASO 3: insertar el 2
* comparar el 2 con el 6 → intercambio
[3,4,2,6,1,0]
* comparar el 2 con el 4 → intercambio
[3,2,4,6,1,0]
* comparar el 2 con el 3 → intercambio
[2,3,4,6,1,0]

PASO 4: insertar el 1
* comparar el 1 con el 4 → intercambio
[2,3,1,4,6,0]
* comparar el 1 con el 3 → intercambio
[2,1,3,4,6,0]
* comparar el 1 con el 2→ intercambio
[1,2,3,4,6,0]

PASO 5 :
* comparar el 0 con el 6 → intercambio
[1,2,3,4,0,6]
* comparar el 0 con el 4 → intercambio
[1,2,3,0,4,6]
* comparar el 0 con el 3 → intercambio
[1,2,0,3,4,6]
* comparar el 0 con el 2 → intercambio
[1,0,2,3,4,6]
* comparar el 0 con el 1 → intercambio
[0,1,2,3,4,6]

Mejor de los casos : que la lista este ordenada
L=[1,3,6,8]

PASO 1: insertar el 3
* comparar el 3 con el 1

PASO 2: insertar el 6

* comparar el 6 con el 3

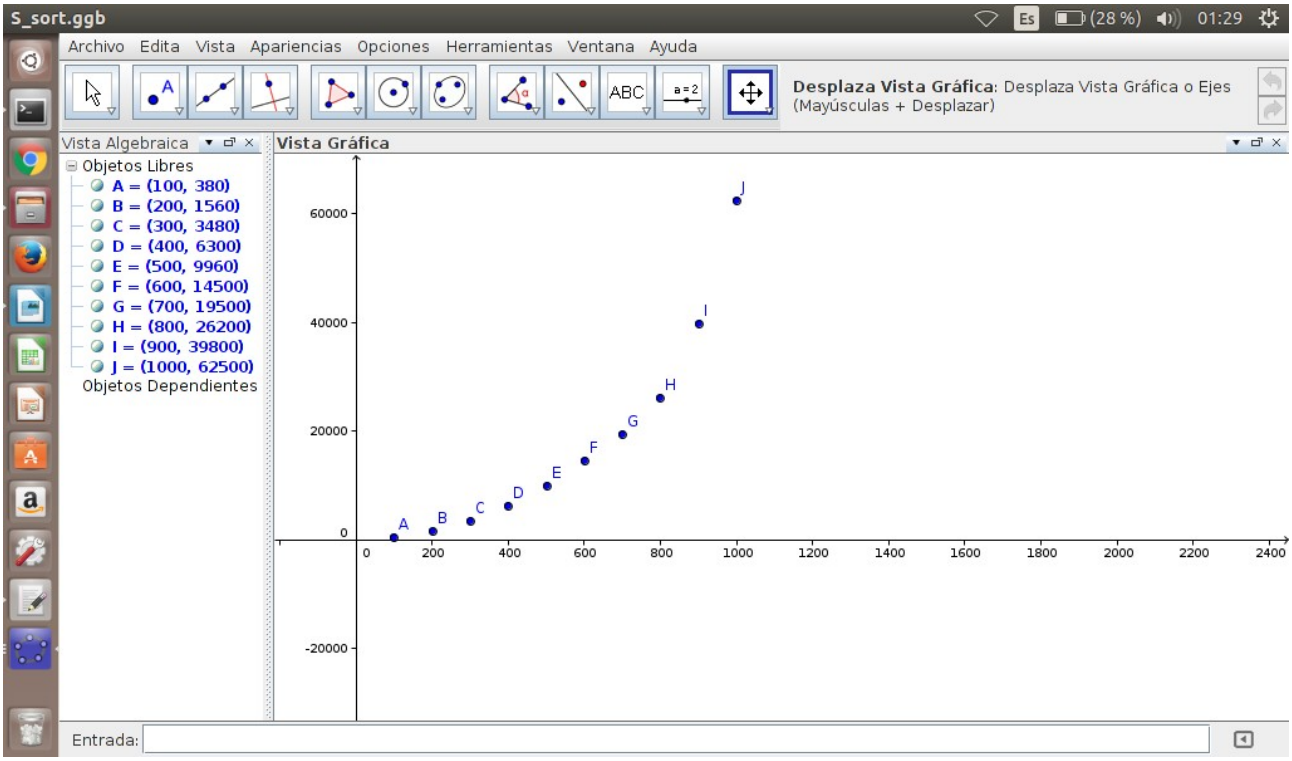
PASO 1: insertar el 8
* comparar el 8 con el 6

5)COMPLEJIDAD DE CADA ALGORITMO:
insercion_sort
°mejor caso: O(n)
°al azar= O(n^2)
°peor caso: O(n^2)

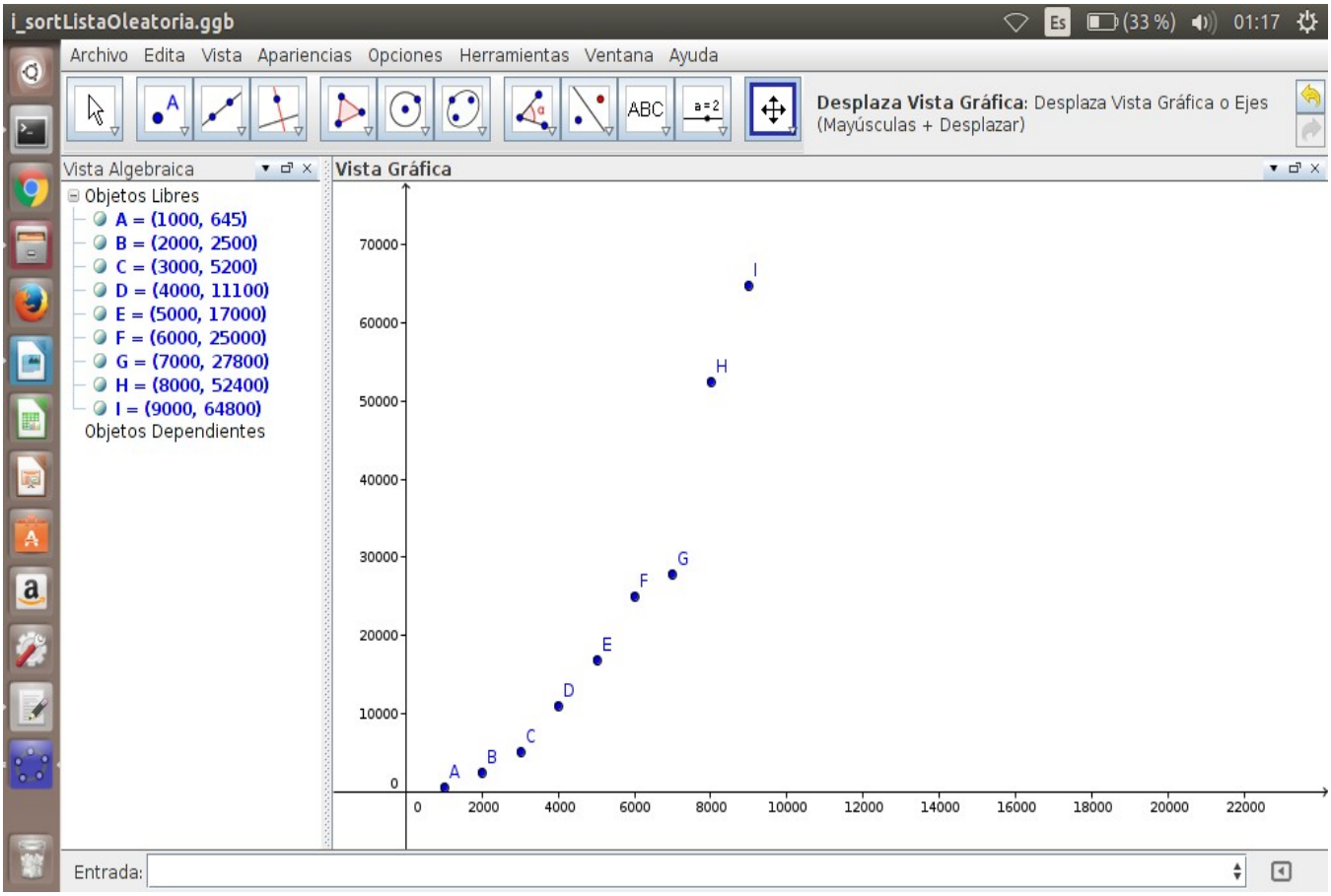
selección_sort
°mejor caso: O(n^2)
°al azar :O(n^2)
°peor caso :O(n^2)

6)Con las mediciones en timeit grafique en geogebra.

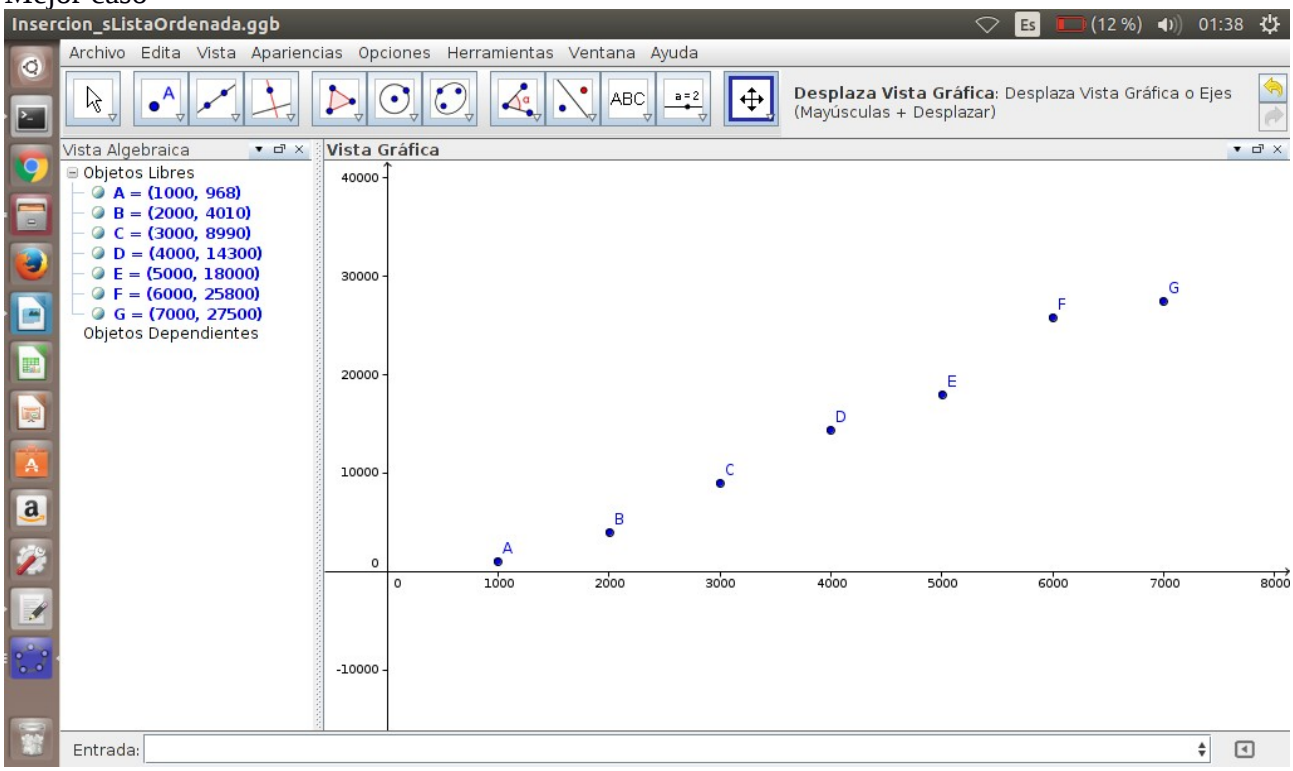
Ordenamiento por Seleccion



Ordenamiento por inserccion
Al azar :



Mejor caso



Peor de los casos