UNIVERSIDADE DA CORUÑA

FACULTAD DE INFORMÁTICA
GRADO EN CIENCIA E INGENIERÍA DE DATOS

Diseño y Análisis de Algoritmos
Práctica 2: Algoritmos de Ordenación

7 de octubre de 2024

Dado un vector de longitud n de números enteros $[a_1, a_2, ..., a_n]$, se pide ordernar sus elementos de menor a mayor, utilizando para ello los algoritmos de ordenación por inserción, burbuja y selección, cuyo pseudocódigo se muestra a continuación:

Insertion sort

```
function insertionSort (v[1..n]):
    for i := 2 to n:
        x := v[i];
        j := i-1;
        while j > 0 and v[j] > x:
            v[j+1]:= v[j];
            j := j-1;
        v[j+1]:= x;
```

Bubble sort

```
function bubbleSort (v[1..n]):
    for i := 2 to n:
        for j := 1 to n-i+1:
            if v[j+1] <v[j]:
                swap(v[j], v[j+1]); #intercambio
```

Selection sort

```
function selectionSort (v[1..n]):
    for i:=1 to n-1:
        minj:=i;
        minx:=v[i];
        for j:=i+1 to n:
            if v[j]<minx:
                minj:=j;
                minx:=v[j];
        v[minj]:=v[i];
        v[i]:=minx;
```

Las tareas a llevar a cabo son:

1. Implementar en PYTHON los algoritmos propuestos.
2. Validar que los algoritmos funcionan correctamente, apoyándose en el statement *assert*. Se pide comprobar al menos las siguientes secuencias:

Input	Result
-9, 4, 13, -1, -5	-9, -5, -1, 4, 13
6, -3, -15, 5, 4, 5, 2	-15, -3, 2, 4, 5, 5, 6
13, 4	4, 13
9	9
7, 6, 6, 5, 4, 3, 2, 1	1, 2, 3, 4, 5, 6, 6, 7
1, 2, 3, 4, 4, 5, 6, 7	1, 2, 3, 4, 4, 5, 6, 7

Así mismo, realizad una segunda comprobación con vectores generados de forma aleatoria para los algoritmos propuestos. La figura 1 muestra un ejemplo con *bubbleSort*.

```python
# generate random integer values
from numpy.random import seed
from numpy.random import randint
import copy

seed(1)

sizes = [5,10,15]

for size in sizes:
        #creates an array with size random elements between -10 and 10
        array = randint(-10,10,size)
        bubbleSort(array)
        #check that after the call to bubbleSort(array) array is correctly sorted
```

Figura 1: Inicialización de un vector con números pseudoaleatorios en el rango $[-10, \ldots, 10]$

3. Determinar los tiempos de ejecución para vectores de n elementos considerando tres situaciones distintas: (a) vectores inicializados aleatoriamente, (b) vectores ordenados de manera ascendente y (c) vectores ordenados de manera descendente. Para los experimentos, escoged una serie de valores de n que se correspondan a una de las progresiones geométricas explicadas en clase de prácticas. Para generar los datos de prueba aleatorios, utilizad como base el código de la figura 1 que genera vectores de números pseudoaleatorios en el rango $[-10, \ldots, 10]$ (se permite adaptar el rango). Para los datos de prueba ordenados podéis consultar el funcionamiento de la función `numpy.arange`.

4. Analizad los resultados obtenidos realizando una comprobación empírica de la complejidad teórica para cada algoritmo y cada una de las situaciones previstas (vectores aleatorios y ordenados de manera creciente y decreciente. Igualmente se realizará una comprobación empírica utilizando un cota subestimada y otra sobrestimada. Los resultados deben presentarse en tablas, de manera similar a lo realizado en la práctica 1.

Evaluación y modo de entrega

- La fecha de entrega límite es el día **19 de octubre de 2024** a las 23:59 horas.
- Será necesario depositar en la página de la asignatura en el CampusVirtual, el fichero Python que contiene el código fuente de la práctica y el informe con el estudio de complejidad.
- Revise detenidamente el documento PlantillaPracticas para saber qué se va a evaluar.
- Solo uno de los integrantes del equipo de prácticas debe depositar el contenido en el CampusVirtual. El nombre de **todos** los integrantes debe figurar en el encabezado de los documentos a entregar.
- Todos los aspectos relacionados con dispensa académica, dedicación al estudio, permanencia y fraude académico se regirán de acuerdo con la normativa académica vigente de la UDC. Si las pruebas o actividades de evaluación se llevan a cabo en grupos, todos los miembros del grupo serán responsables solidariamente por el trabajo realizado y entregado, así como de sus posibles consecuencias..