



## **TALLER MÉTODOS DE ORDENAMIENTO**

### **Presenta**

Cristian David Mora Saenz

### **Docente**

Segundo Fidel Puerto Garavito

### **Asignatura**

Diseño de Algoritmos

NRC:

7487

Corporación Universitaria Minuto de Dios

Facultad de Ingeniería

Bogotá D.C, Colombia      febrero 28 de 2020.

## **Introducción:**

Para este ejercicio ingresaremos en un archivo datos de diferentes magnitudes, el archivo utilizara una cantidad total de 100 (cien), 1000 (mil), 10000 (diez mil), 100000 (cien mil), 1000000 (un millón), todos los datos ingresados a este archivo serán de tipo numérico y estarán separados por comas.

El ejercicio mostrara el tiempo que se ha tardado el programa en ser ejecutado, además de la forma en la que los datos no están ordenados y posteriormente la forma ordenada de los datos.

El ejercicio implementara tres métodos de ordenamiento;

1. Método de Ordenamiento Burbuja.
2. Método de Ordenamiento QuickSort u Ordenamiento rápido
3. Método de Ordenamiento por Inserción.

**NOTA:** Las imágenes, fueron tomadas con ayuda del compilador online de Python:

<https://repl.it/languages/python3>, sin embargo, las imágenes a partir de los 1000 datos están recortadas y solo se muestra una fracción de los datos, debido a la cantidad de número que no podían ser visualizados en una sola imagen.

## Desarrollo:

1. **Método de Ordenamiento Burbuja.** Este método de ordenamiento consiste en realizar comparaciones entre los datos almacenados dentro de un *Arreglo* estas comparaciones se realizan de forma secuencial, lo que hace que este método de ordenamiento sea lento o quizás nunca termine su ejecución.

Código:

```
import random # Importamos la clase random para los datos randomicos.
import time # Se importa la clase time, para saber el tiempo de ejecucion del programa.

# El siguiente método es el ordenamiento burbuja y recibe como parametro la lista que conforma el archivo.
def ordenamientoBurbuja(unalista):
    # Los siguientes ciclos for, se implementan para recorrer el arreglo, que contiene los datos
    # almacenados por el archivo.
    for numPasada in range(len(unalista)-1,0,-1):
        for i in range(numPasada):
            # El condicional if se utiliza para realizar las comparaciones entre los datos.
            if unalista[i]>unalista[i+1]:
                temp = unalista[i]
                unalista[i] = unalista[i+1]
                unalista[i+1] = temp

#Escribe numeros en un archivo llamado "Ejemplo.txt"

File = open("Ejemplo.txt", "w")
for i in range(100): # Se utiliza el ciclo for, para llenar el archivo con los datos randomicos
    # y separados por comas.
    File.write( str(random.randint(1,1000)) + "," ) # La función Write se utiliza para escribir sobre el archivo,
    # teniendo en cuenta que estan separados por comas.

File.close() # La función Close, se utiliza para cerrar el archivo sobre el cual se esta escribiendo.

# Leer y sumar los numeros del archivo "Ejemplo.txt"
fr = open("Ejemplo.txt", "r")
for line in fr: # El ciclo for en este caso, se utiliza para leer los datos y las lineas del archivo.
    files = line.split(",") # El Método Split se encarga de leer solamente los datos, que no sean comas (,).

tam = len(files)-1 # Len sirve para obtener la longitud.
```

```
# Impreción de los arreglos ordenados y no ordenados.
print ("No Ordenado \n")

print(files)

print ("Ordenado \n")
ordenamientoBurbuja(files)
print(files)

hms = time.time()

# La siguiente linea expresa el tiempo de ejecucion
print((time.time()) - hms)

fr.close()
```

- Ejecución: 100 Datos (Cien)

```
No Ordenado
['383', '765', '287', '217', '438', '546', '84', '704', '501',
'319', '355', '978', '102', '737', '459', '101', '680', '386',
'80', '369', '368', '551', '776', '170', '209', '463', '559', '
783', '49', '527', '916', '558', '290', '402', '897', '712', '8
12', '436', '958', '901', '837', '670', '447', '424', '565', '5
82', '681', '44', '960', '573', '373', '659', '969', '564', '94
8', '645', '492', '814', '631', '240', '678', '972', '964', '62
9', '581', '314', '934', '223', '825', '191', '670', '675', '81
2', '262', '500', '163', '709', '323', '144', '34', '546', '395
', '455', '6', '799', '114', '301', '64', '438', '539', '345',
'254', '896', '355', '997', '870', '800', '697', '256', '54', '
']

Ordenado
['', '101', '102', '114', '144', '163', '170', '191', '209', '2
17', '223', '240', '254', '256', '262', '287', '290', '301', '3
14', '319', '323', '34', '345', '355', '355', '368', '369', '37
3', '383', '386', '395', '402', '424', '436', '438', '438', '44
', '447', '455', '459', '463', '49', '492', '500', '501', '527'
, '539', '54', '546', '546', '551', '558', '559', '564', '565',
'573', '581', '582', '6', '629', '631', '64', '645', '659', '6
70', '670', '675', '678', '680', '681', '697', '704', '709', '7
12', '737', '765', '776', '783', '799', '80', '800', '812', '81
2', '814', '825', '837', '84', '870', '896', '897', '901', '916
', '934', '948', '958', '960', '964', '969', '972', '978', '997
']
```

- Tiempo de ejecución: 6.532669067382812e-05 Segundos

- Ejecución: 1000 Datos (Mil)

```
Ordenado
['', '1', '1', '100', '100', '100', '100', '1000', '1000', '101',
 '101', '107', '108', '109', '111', '112', '113', '113', '114',
 '114', '118', '118', '12', '12', '120', '121', '122', '123',
 '123', '125', '128', '129', '130', '131', '132', '133', '134',
 '136', '137', '138', '139', '14', '140', '142', '142', '143',
 '144', '145', '145', '146', '146', '146', '146', '148', '149',
 '149', '149', '149', '15', '15', '15', '152', '152', '152', '15',
 '154', '154', '156', '156', '156', '156', '159', '160', '16',
 '161', '161', '162', '163', '165', '167', '168', '168', '16',
 '17', '170', '170', '172', '173', '173', '174', '175', '175',
 '175', '178', '179', '18', '181', '182', '182', '184', '186',
 '189', '19', '19', '19', '19', '19', '191', '191', '192', '192', '1',
 '192', '194', '194', '194', '198', '198', '198', '199', '2', '2',
 '20', '20', '20', '200', '202', '204', '205', '206', '206', '2',
 '207', '207', '208', '208', '210', '210', '211', '212', '212', '2',
 '212', '213', '213', '214', '215', '216', '216', '216', '217', '2',
 '217', '217', '218', '219', '219', '22', '220', '220', '225', '22',
 '227', '227', '231', '232', '235', '235', '236', '24', '24',
 '240', '240', '242', '243', '243', '245', '245', '247', '249',
 '25', '250', '250', '251', '253', '253', '255', '261', '263',
 '263', '264', '266', '268', '268', '269', '269', '27', '272',
 '272', '273', '274', '275', '275', '276', '276', '278', '279',
 '279', '281', '281', '282', '284', '288', '289', '29', '290',
 '291', '292', '292', '293', '293', '294', '294', '294', '295',
 '295', '296', '298', '298', '298', '30', '301', '302', '303', '3',
 '306', '307', '307', '308', '308', '31', '311', '311', '312', '31',
 '312', '314', '316', '316', '316', '317', '318', '32', '323',
 '323', '323', '323', '324', '325', '328', '328', '329', '329',
 '329', '33', '33', '332', '332', '332', '333', '334', '337',
 '34', '34', '340', '341', '342', '342', '342', '343', '343',
 '344', '346', '348', '350', '350', '350', '355', '356', '3
```

- Tiempo de ejecución: 7.939338684082031e-05 Segundos

- Ejecución: 10000 Datos (Diez Mil)

```
Ordenado
['', '1', '10', '100', '101', '101', '102', '104', '105', '107',
, '108', '109', '109', '109', '111', '111', '112', '113', '115',
, '115', '115', '118', '119', '119', '119', '119', '119', '122',
, '123', '126', '127', '127', '127', '127', '128', '13', '13',
'130', '130', '132', '133', '135', '136', '136', '136', '138',
'139', '140', '140', '141', '144', '146', '146', '147', '148',
'15', '150', '150', '153', '154', '155', '156', '157', '161', '
162', '162', '163', '163', '164', '164', '164', '165', '166', '
168', '17', '171', '171', '174', '174', '174', '174', '175', '1
76', '177', '178', '180', '180', '180', '181', '181', '181', '1
82', '182', '182', '182', '183', '183', '185', '186', '186', '1
88', '188', '189', '19', '190', '193', '194', '194', '194', '19
5', '196', '196', '197', '199', '2', '2', '200', '200', '200',
'201', '202', '207', '207', '207', '207', '208', '209', '21', '
211', '212', '214', '215', '216', '216', '217', '218', '219', '
22', '221', '222', '223', '225', '226', '227', '227', '228', '2
28', '231', '232', '232', '232', '233', '235', '236', '236', '2
36', '237', '237', '237', '237', '237', '24', '24', '240', '242
', '243', '244', '245', '245', '246', '246', '247', '247', '250
', '252', '253', '254', '254', '256', '256', '256', '257', '257
', '258', '259', '259', '259', '259', '26', '26', '260', '262',
'265', '266', '267', '268', '269', '271', '271', '271', '273',
'276', '276', '277', '278', '28', '280', '280', '281', '283',
'284', '286', '289', '289', '29', '292', '292', '293', '294', '
295', '296', '296', '298', '299', '3', '3', '3', '3', '30', '30
0', '300', '300', '302', '303', '304', '305', '305', '305', '30
6', '306', '307', '31', '311', '312', '313', '315', '315', '316
', '317', '318', '319', '320', '321', '323', '324', '326', '327
', '327', '327', '328', '33', '33', '330', '333', '333', '334',
'334', '337', '338', '339', '34', '34', '340', '340', '341', '
344', '345', '345', '346', '35', '350', '350', '351', '351', '3
51', '353', '353', '354', '354', '355', '355', '356', '356', '3
56', '357', '357', '358', '359', '361', '361', '364', '36', '3
```

- Tiempo de ejecución: 9.369850158691406e-05 Segundos

**NOTA:** El método de ordenamiento Burbuja, no es capaz de terminar el programa con más de 100000 (Cien mil) y 1000000 (un millón) de datos, debido a la característica que posee con respecto a la comparación entre cada dato y a la secuencialidad del método, el proceso es muy tardío.

2. **Método de ordenamiento QuickSort:** es un algoritmo basado en la técnica de divide y vencerás, que permite, en promedio, ordenar  $n$  elementos en un tiempo proporcional a  $n \log n$ .

- Código:

```
import random # Importamos la clase random para los datos randomicos.
import time # Se importa la clase time, para saber el tiempo de ejecucion del programa.

# El siguiente es el método de ordenamiento QuickSort, acepta como parametro una lista (array), la cual es la
# Traducción de nuestros datos, almacenados por comas en un archivo.
def QuickSort(lista):
    # Las siguientes 3 listas, se utilizan para recorrer este arreglo y realizar las comparaciones.
    izquierda = []
    centro = []
    derecha = []
    # El siguiente condicional depende si nuestro arreglo contiene elementos.
    if len(lista) > 1:
        # La variable "pivote" se inicializa en el primer dato del array, y se utiliza para tener de ejemplo
        # ese dato, para compararlo con los demas, hasta hallar uno menor, igual o mayor:
        """
        1. Si el dato es menor se convertira en el nuevo pivote-
        2. Si el dato es igual, se coloca despues del pivote.
        3. Si el dato es mayor, debe ir despues del pivote.
        """
        pivote = lista[0]
        for i in lista: # Este ciclo for, recorre el array.
            # los siguiente condicionales realizan las comparaciones con respecto a nuestro pivote, con respecto
            # a los demas elementos que componen el array.
            if i < pivote:
                izquierda.append(i)
            elif i == pivote:
                centro.append(i)
            elif i > pivote:
                derecha.append(i)
        #print(izquierda+["-"]+centro+["-"]+derecha)
        return QuickSort(izquierda)+centro+QuickSort(derecha)
    else:
        return lista
```

```
#Escribe numeros en un archivo llamado "Ejemplo.txt"

File = open("Ejemplo.txt", "w")
for i in range(100): # Se utiliza el ciclo for, para llenar el archivo con los datos randomicos
    # y separados por comas.
    File.write( str(random.randint(1,1000)) + "," ) # La función Write se utiliza para escribir sobre el archivo,
    # teniendo en cuenta que estan separados por comas.

File.close() # La función Close, se utiliza para cerrar el archivo sobre el cual se esta escribiendo.

# Leer y sumar los numeros del archivo "Ejemplo.txt"
fr = open("Ejemplo.txt", "r")
for line in fr: # El ciclo for en este caso, se utiliza para leer los datos y las lineas del archivo.
    files = line.split(",") # El Método Split se encarga de leer solamente los datos, que no sean comas (,).

tam = len(files)-1 # Len sirve para obtener la longitud.

# Impreción de los arreglos ordenados y no ordenados.
print ("No Ordenado \n")

print(files)

print ("Ordenado \n")

print(QuickSort(files))

hms = time.time()

# La siguiente linea expresa el tiempo de ejecucion
print((time.time()) - hms)

fr.close()
```

- Ejecución: 100 Datos

No ordenados:

```
No Ordenado

['885', '437', '753', '467', '243', '569', '629', '91', '316', '452', '293', '150', '777', '712',
'853', '68', '595', '799', '364', '175', '891', '381', '315', '54', '638', '92', '16', '227', '610',
', '680', '912', '181', '727', '575', '25', '985', '182', '513', '262', '164', '763', '727', '300',
', '617', '927', '572', '763', '866', '934', '207', '62', '468', '210', '392', '761', '322', '929',
', '28', '776', '944', '800', '219', '948', '183', '628', '7', '592', '356', '624', '506', '988', '2',
55', '291', '78', '817', '316', '107', '812', '871', '949', '724', '582', '653', '858', '518', '68',
1', '86', '395', '153', '330', '988', '774', '830', '254', '167', '393', '663', '949', '668', '871',
', '']
```

Ordenado:

```
Ordenado

['', '107', '150', '153', '16', '164', '167', '175', '181', '182', '183', '207', '210', '219', '22',
7', '243', '25', '254', '255', '262', '28', '291', '293', '300', '315', '316', '316', '322', '330',
', '356', '364', '381', '392', '393', '395', '437', '452', '467', '468', '506', '513', '518', '54',
', '569', '572', '575', '582', '592', '595', '610', '617', '62', '624', '628', '629', '638', '653',
', '663', '668', '68', '680', '681', '7', '712', '724', '727', '727', '753', '761', '763', '763', '77',
4', '776', '777', '78', '799', '800', '812', '817', '830', '853', '858', '86', '866', '871', '871',
', '885', '891', '91', '912', '92', '927', '929', '934', '944', '948', '949', '949', '985', '988',
', '988']
```

- Tiempo de ejecución: **1.1920928955078125e-06** Segundos



- Ejecución: 1000 Datos

No Ordenado:

```
No Ordenado

['856', '711', '318', '906', '913', '107', '578', '394', '598', '179', '276', '961', '668', '452',
 '233', '632', '427', '729', '770', '352', '896', '497', '969', '850', '480', '962', '888', '815',
 '816', '299', '477', '524', '184', '907', '862', '119', '921', '858', '642', '977', '676', '695',
 '918', '361', '992', '304', '728', '10', '906', '835', '203', '27', '955', '172', '574', '898',
 982', '564', '858', '600', '214', '367', '414', '460', '680', '363', '221', '784', '587', '939',
 475', '250', '16', '800', '504', '27', '196', '181', '657', '104', '659', '369', '974', '943', '77
 8', '225', '407', '766', '935', '586', '24', '707', '842', '718', '103', '853', '151', '637', '565
 ', '310', '385', '464', '706', '180', '562', '552', '922', '768', '59', '843', '939', '260', '441'
 ', '418', '381', '553', '472', '600', '319', '753', '180', '256', '198', '240', '513', '714', '379'
 ', '716', '912', '313', '135', '199', '310', '888', '624', '446', '461', '566', '710', '356', '339'
 ', '857', '321', '409', '146', '385', '150', '800', '834', '953', '652', '613', '656', '253', '174'
 ', '635', '571', '896', '535', '28', '501', '207', '888', '133', '22', '242', '808', '569', '10',
 574', '59', '719', '642', '842', '948', '634', '555', '233', '926', '63', '325', '618', '167', '61
 5', '709', '771', '817', '503', '59', '634', '130', '813', '558', '522', '972', '352', '172', '142
 ', '563', '290', '748', '592', '154', '771', '683', '895', '882', '890', '661', '119', '416', '760
 ', '815', '464', '571', '809', '241', '498', '571', '369', '920', '348', '254', '224', '754', '168
 ', '930', '670', '708', '637', '977', '968', '118', '915', '513', '890', '10', '189', '563', '624'
 ', '729', '224', '995', '57', '316', '253', '846', '664', '866', '707', '899', '128', '539', '378',
 ', '233', '858', '748', '293', '607', '837', '744', '89', '223', '933', '351', '464', '268', '699',
 ', '837', '560', '354', '204', '381', '909', '325', '777', '706', '955', '402', '85', '543', '680',
 250', '422', '140', '494', '777', '279', '752', '734', '235', '366', '886', '721', '929', '300',
 323', '522', '485', '684', '327', '211', '200', '972', '420', '993', '548', '458', '860', '317',
 186', '10', '947', '566', '301', '985', '374', '799', '650', '393', '488', '210', '394', '220', '9
 42', '18', '841', '12', '448', '815', '918', '485', '742', '414', '332', '980', '799', '351', '177
 ', '928', '531', '863', '826', '137', '821', '637', '378', '717', '939', '80', '973', '642', '391'
 ', '155', '417', '711', '751', '5', '637', '762', '317', '395', '63', '948', '795', '469', '936',
 667', '711', '732', '459', '883', '752', '55', '736', '325', '603', '691', '674', '547', '896', '6
 95', '952', '739', '765', '725', '475', '858', '773', '663', '101', '276', '85', '226', '219', '22
```

Ordenado:

```
Ordenado

['', '10', '10', '10', '10', '100', '101', '101', '102', '102', '103', '104', '104', '107', '11',
 '112', '114', '117', '118', '119', '119', '119', '12', '12', '12', '120', '121', '122', '126', '12
 7', '127', '128', '13', '13', '130', '132', '133', '135', '136', '136', '137', '140', '141', '142'
 ', '144', '144', '146', '147', '147', '15', '150', '150', '151', '154', '155', '158', '159', '159',
 '16', '160', '161', '161', '163', '163', '167', '167', '168', '168', '168', '17', '172', '172', '173',
 174', '174', '177', '179', '18', '18', '180', '180', '181', '181', '182', '184', '184', '186', '18
 7', '188', '189', '192', '195', '196', '197', '198', '199', '200', '200', '201', '202', '203', '20
 4', '204', '205', '206', '206', '207', '208', '21', '210', '211', '214', '216', '217', '218', '219
 ', '22', '22', '220', '220', '221', '221', '221', '221', '222', '223', '224', '224', '225', '225',
 ', '226', '226', '229', '23', '23', '230', '232', '232', '233', '233', '233', '234', '234', '235',
 236', '237', '24', '240', '241', '242', '242', '244', '246', '247', '247', '249', '25', '250', '25
 0', '252', '253', '253', '254', '255', '255', '256', '256', '256', '256', '260', '261', '263', '265', '26
 8', '27', '27', '27', '271', '272', '273', '274', '276', '276', '278', '279', '28', '28', '28', '2
 80', '281', '281', '282', '283', '283', '286', '288', '29', '290', '292', '293', '293', '293', '29
 3', '294', '298', '299', '300', '301', '304', '305', '306', '306', '307', '307', '310', '310', '31
 3', '313', '315', '316', '317', '317', '318', '318', '319', '32', '32', '321', '321', '322', '322'
 ', '323', '323', '325', '325', '325', '326', '326', '327', '327', '328', '330', '332', '332', '334'
 ', '336', '336', '336', '337', '339', '341', '343', '344', '346', '348', '35', '350', '351', '351',
 ', '351', '352', '352', '352', '354', '355', '355', '356', '357', '357', '358', '359', '361', '362',
 '363', '366', '367', '369', '369', '369', '370', '370', '371', '373', '374', '376', '378', '378',
 '378', '379', '379', '381', '381', '382', '383', '384', '385', '385', '385', '386', '388', '388',
 '388', '390', '390', '390', '391', '393', '393', '394', '394', '395', '397', '4', '40', '40', '40
 ', '401', '401', '401', '401', '402', '402', '402', '404', '407', '407', '408', '408', '408', '409
 ', '41', '41', '411', '411', '412', '414', '414', '416', '416', '417', '417', '418', '420', '422',
 '422', '427', '429', '430', '434', '434', '435', '435', '437', '437', '439', '44', '441', '443',
 '446', '446', '446', '447', '447', '447', '448', '449', '451', '452', '452', '452', '454', '454',
 '458', '459', '460', '460', '461', '461', '461', '461', '464', '464', '464', '465', '467', '468',
 '469', '469', '471', '472', '472', '475', '475', '475', '477', '479', '48', '480', '481', '484',
 485', '485', '485', '486', '487', '488', '488', '489', '490', '491', '491', '494', '495', '496',
 497', '497', '498', '499', '499', '5', '5', '50', '500', '501', '502', '503', '503', '503', '504',
```

- Tiempo de Ejecución: 3.0994415283203125e-06 Segundos

- Ejecución: 10000 Datos (Diez Mil)

No Ordenado:

```
No Ordenado
['374', '780', '386', '847', '487', '33', '316', '470', '747', '561', '1', '369', '309', '
11', '149', '459', '651', '408', '106', '839', '532', '421', '524', '663', '276', '639', '
217', '110', '952', '584', '14', '620', '351', '663', '83', '281', '200', '502', '722', '
118', '583', '760', '990', '794', '455', '296', '278', '346', '157', '823', '35', '752', '
605', '765', '187', '334', '949', '653', '301', '243', '719', '115', '747', '149', '354
', '764', '576', '986', '953', '658', '70', '557', '442', '766', '488', '840', '734', '68
82', '404', '917', '1000', '755', '295', '328', '236', '417', '386', '392', '461', '607', '
999', '873', '793', '842', '618', '267', '931', '290', '830', '648', '657', '164', '766
42', '572', '804', '625', '514', '549', '286', '17', '794', '801', '79', '741', '708', '
667', '113', '223', '417', '752', '244', '149', '994', '626', '217', '9', '38', '266', '3
', '850', '647', '120', '435', '72', '308', '8', '625', '513', '211', '592', '658', '822'
', '939', '237', '325', '3', '817', '153', '168', '74', '28', '237', '627', '301', '953',
', '108', '506', '194', '206', '697', '28', '396', '355', '941', '141', '356', '351', '54',
', '148', '656', '753', '484', '634', '601', '746', '920', '782', '168', '366', '322', '38
', '784', '663', '87', '13', '203', '746', '7', '933', '125', '696', '362', '194', '933',
', '958', '503', '515', '572', '420', '263', '308', '507', '747', '156', '671', '482', '763'
', '985', '154', '610', '857', '779', '881', '588', '967', '149', '530', '459', '648', '29
9', '885', '7', '399', '278', '55', '450', '478', '454', '764', '388', '773', '759', '775
', '127', '444', '606', '35', '6', '134', '967', '343', '245', '536', '78', '225', '876',
', '87', '470', '544', '173', '936', '617', '421', '143', '194', '113', '80', '387', '743',
', '465', '562', '643', '799', '771', '995', '282', '801', '674', '665', '965', '663', '280
', '593', '834', '36', '561', '185', '483', '135', '684', '220', '317', '280', '942', '24
73', '331', '335', '524', '111', '462', '415', '249', '300', '479', '723', '88', '811', '
68', '543', '951', '669', '321', '619', '607', '600', '96', '853', '451', '541', '685', '
62', '872', '509', '89', '296', '202', '489', '947', '316', '45', '135', '219', '345', '9
', '438', '627', '704', '9', '44', '67', '969', '622', '97', '891', '288', '88', '713', '
84', '570', '693', '193', '935', '961', '203', '839', '722', '604', '15', '687', '41', '1
', '427', '306', '274', '633', '30', '741', '638', '584', '932', '222', '225', '892', '249
', '149', '128', '155', '127', '126', '168', '152', '127', '121', '151', '163', '175', '163', '175', '166'
```

Ordenado:

```
Ordenado
['', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '10', '10', '10', '10', '10',
, '100', '100', '100', '100', '100', '100', '100', '100', '100', '100', '100', '100', '1000', '1000',
, '1000', '101', '101', '101', '101', '101', '101', '101', '101', '101', '101', '101', '101', '101', '1
102', '102', '102', '103', '103', '103', '103', '103', '103', '103', '103', '103', '103', '103', '103',
, '105', '105', '105', '105', '105', '105', '105', '105', '105', '105', '105', '105', '105', '105', '10
06', '106', '106', '106', '106', '107', '107', '107', '107', '107', '107', '107', '107', '107', '107',
, '108', '108', '108', '108', '108', '108', '108', '108', '109', '109', '109', '109', '109', '109', '109
, '11', '11', '11', '11', '11', '110', '110', '110', '110', '110', '110', '110', '110', '110', '1
111', '111', '111', '111', '111', '111', '111', '111', '111', '111', '111', '111', '111', '111',
, '113', '113', '113', '113', '113', '113', '113', '113', '113', '113', '113', '113', '114', '114', '11
15', '115', '115', '115', '116', '116', '116', '116', '116', '116', '116', '117', '117', '117',
, '117', '117', '117', '117', '117', '117', '118', '118', '118', '118', '118', '118', '118', '118', '118
9', '119', '119', '119', '119', '119', '119', '119', '119', '119', '12', '12', '12', '12', '12', '12',
, '120', '120', '120', '120', '120', '120', '121', '121', '121', '121', '122', '122', '122', '12
23', '123', '123', '123', '123', '123', '123', '123', '123', '124', '124', '124', '124', '124', '124',
, '125', '125', '125', '125', '126', '126', '126', '126', '126', '126', '126', '126', '126', '126', '126
7', '127', '127', '128', '128', '128', '128', '129', '129', '129', '129', '129', '129', '129', '129', '
, '13', '13', '13', '13', '13', '13', '13', '13', '13', '13', '130', '130', '130', '130', '130', '130',
, '131', '131', '131', '131', '131', '131', '131', '131', '131', '132', '132', '132', '132', '132', '13
33', '133', '133', '133', '133', '134', '134', '134', '134', '134', '134', '134', '134', '134', '134',
, '135', '135', '135', '135', '135', '135', '136', '136', '136', '136', '136', '136', '136', '136', '136
7', '137', '137', '137', '137', '137', '138', '138', '138', '138', '138', '138', '138', '138', '138', '
, '139', '139', '139', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '1
0', '140', '140', '140', '140', '140', '140', '141', '141', '141', '141', '141', '141', '141', '141', '
, '142', '142', '142', '142', '142', '142', '142', '142', '142', '142', '143', '143', '143', '143',
, '144', '144', '144', '144', '144', '144', '144', '145', '145', '145', '145', '145', '145', '145', '1
147', '147', '147', '147', '147', '147', '147', '147', '147', '147', '148', '148', '148', '148', '148
, '148', '148', '148', '148', '149', '149', '149', '149', '149', '149', '149', '149', '149', '149', '14
, '150', '150', '150', '150', '150', '150', '150', '150', '151', '151', '151', '151', '151', '151', '1
152', '152', '153', '153', '153', '153', '153', '153', '153', '153', '153', '153', '153', '153', '153',
, '154', '154', '154', '154', '154', '154', '154', '154', '154', '154', '154', '154', '154', '154', '15
```

- Tiempo de Ejecución: **4.291534423828125e-06** Segundos

'156'	'437'	'518'	'356'	'924'	'330'	'898'	'97'	'85'	'219'	'857'	'223'	'25'
'666'	'551'	'544'	'791'	'78'	'922'	'863'	'316'	'431'	'33'	'501'	'356'	'256'
'445'	'7'	'989'	'767'	'284'	'559'	'967'	'818'	'846'	'869'	'619'	'636'	'180'
'731'	'606'	'777'	'355'	'564'	'478'	'730'	'633'	'379'	'420'	'597'	'757'	'447'
'330'	'405'	'852'	'202'	'567'	'508'	'997'	'216'	'831'	'772'	'489'	'9'	'365'
'252'	'214'	'433'	'315'	'576'	'647'	'397'	'707'	'990'	'616'	'534'	'71'	'70'
'469'	'590'	'875'	'631'	'409'	'845'	'258'	'366'	'140'	'728'	'252'	'479'	'3'
'782'	'923'	'523'	'618'	'581'	'460'	'249'	'795'	'516'	'993'	'111'	'328'	'623'
'806'	'799'	'529'	'582'	'275'	'487'	'559'	'91'	'575'	'153'	'208'	'913'	'542'
'333'	'558'	'446'	'798'	'971'	'30'	'868'	'515'	'547'	'706'	'540'	'860'	'90'
'364'	'334'	'11'	'803'	'10'	'734'	'706'	'753'	'820'	'426'	'307'	'768'	'619'
'32'	'646'	'68'	'227'	'882'	'653'	'928'	'710'	'334'	'890'	'828'	'997'	'787'
'818'	'673'	'415'	'883'	'955'	'561'	'808'	'942'	'907'	'689'	'880'	'129'	'18'
'6'	'265'	'34'	'778'	'114'	'857'	'83'	'762'	'155'	'452'	'255'	'430'	'74'
'613'	'953'	'492'	'42'	'927'	'462'	'471'	'902'	'534'	'297'	'846'	'723'	'54'
'6'	'613'	'497'	'8'	'250'	'491'	'606'	'808'	'442'	'209'	'763'	'197'	'976'
'0'	'334'	'850'	'650'	'359'	'87'	'28'	'935'	'449'	'461'	'933'	'708'	'70'
'953'	'152'	'293'	'1000'	'488'	'291'	'523'	'77'	'839'	'765'	'222'	'386'	'5'
'74'	'207'	'412'	'666'	'169'	'965'	'93'	'180'	'582'	'679'	'372'	'257'	'219'
'168'	'896'	'563'	'506'	'9'	'609'	'447'	'365'	'536'	'533'	'460'	'185'	'202'
'59'	'801'	'450'	'54'	'475'	'781'	'591'	'715'	'671'	'233'	'810'	'511'	'333'
'964'	'506'	'936'	'761'	'871'	'146'	'828'	'61'	'53'	'823'	'678'	'54'	'241'
'55'	'204'	'184'	'457'	'852'	'103'	'570'	'254'	'276'	'483'	'175'	'987'	'218'
'57'	'259'	'916'	'114'	'432'	'890'	'392'	'473'	'304'	'596'	'722'	'894'	'352'
'766'	'782'	'744'	'950'	'231'	'71'	'5'	'876'	'77'	'249'	'561'	'957'	'84'
'31'	'468'	'181'	'469'	'648'	'251'	'557'	'100'	'337'	'315'	'814'	'349'	'32'
'453'	'728'	'365'	'975'	'197'	'778'	'241'	'638'	'245'	'440'	'955'	'736'	'691'
'524'	'315'	'576'	'513'	'68'	'329'	'289'	'158'	'4'	'653'	'803'	'588'	'672'
'25'	'693'	'167'	'969'	'794'	'938'	'574'	'441'	'173'	'35'	'15'	'597'	'446'
'264'	'402'	'366'	'642'	'963'	'368'	'933'	'943'	'618'	'712'	'512'	'198'	'398'
'587'	'957'	'603'	'392'	'280'	'3'	'745'	'370'	'427'	'844'	'677'	'729'	'166'
'35'	'742'	'272'	'119'	'714'	'429'	'525'	'461'	'58'	'89'	'327'	'20'	'388'
'306'	'709'	'941'	'998'	'781'	'453'	'475'	'382'	'61'	'100'	'697'	'158'	'562'
'189'	'552'	'184'	'884'	'438'	'181'	'473'	'364'	'382'	'852'	'523'	'552'	'337'

[illegible]

- Tiempo de Ejecución: 4.291534423828125e-06 Segundos



- Ejecución: 1000000 (Un millón) de Datos

No ordenados:

```
, '65', '151', '833', '676', '529', '393', '358', '423', '680', '379', '883', '339', '282',  
, '386', '136', '638', '865', '439', '76', '531', '308', '346', '791', '147', '79', '949',  
, '548', '949', '823', '596', '964', '27', '585', '15', '859', '579', '205', '218', '676',  
, '161', '249', '174', '217', '631', '607', '151', '929', '388', '405', '364', '43', '81',  
, '2', '616', '767', '582', '447', '702', '639', '666', '563', '27', '378', '398', '668', '8',  
, '07', '301', '717', '492', '961', '189', '508', '293', '15', '386', '666', '724', '816', '1',  
, '571', '338', '380', '92', '934', '919', '282', '155', '406', '175', '350', '494', '977', '1',  
, '154', '290', '409', '840', '959', '852', '629', '606', '343', '68', '693', '257', '188', '1',  
, '118', '799', '145', '153', '186', '419', '860', '294', '442', '153', '487', '89', '890', '1',  
, '178', '328', '868', '198', '676', '714', '317', '342', '355', '663', '15', '977', '17', '1',  
, '53', '762', '956', '480', '175', '465', '151', '88', '141', '106', '605', '767', '794', '1',  
, '43', '824', '116', '243', '862', '508', '494', '603', '585', '273', '209', '813', '940', '1',  
, '720', '558', '569', '323', '325', '597', '853', '676', '156', '462', '279', '77', '54', '1',  
, '7', '550', '365', '473', '121', '732', '82', '709', '728', '662', '47', '19', '402', '748', '1',  
, '604', '70', '557', '625', '475', '973', '527', '907', '777', '743', '101', '11', '137', '1',  
, '155', '202', '688', '644', '621', '270', '452', '868', '21', '517', '968', '623', '88', '1',  
, '9', '867', '855', '680', '329', '96', '353', '217', '605', '138', '705', '66', '527', '67', '1',  
, '0', '337', '380', '379', '937', '263', '126', '323', '632', '515', '708', '858', '279', '1',  
, '452', '141', '102', '426', '582', '271', '133', '988', '746', '614', '607', '915', '335', '1',  
, '703', '190', '807', '394', '459', '475', '349', '351', '935', '435', '807', '273', '809', '1',  
, '166', '582', '505', '75', '143', '331', '975', '556', '422', '157', '666', '820', '83', '1',  
, '0', '106', '343', '438', '32', '696', '178', '799', '609', '215', '933', '12', '702', '93', '1',  
, '4', '473', '939', '158', '275', '430', '793', '786', '511', '378', '806', '569', '802', '1',  
, '350', '914', '720', '558', '19', '995', '348', '941', '471', '137', '15', '915', '265', '1',  
, '153', '795', '503', '376', '429', '266', '420', '432', '558', '455', '335', '702', '133', '1',  
, '931', '522', '235', '365', '585', '778', '206', '533', '924', '108', '53', '587', '968', '1',  
, '896', '538', '764', '107', '438', '564', '170', '117', '819', '944', '25', '749', '573', '1',  
, '745', '956', '690', '614', '341', '520', '657', '850', '713', '439', '794', '5', '925', '1',  
, '360', '613', '456', '742', '461', '346', '460', '341', '639', '314', '37', '514', '28', '1',  
, '2', '77', '179', '42', '557', '804', '765', '937', '648', '328', '822', '276', '817', '54', '1',  
, '7', '682', '8', '57', '668', '517', '766', '209', '715', '466', '328', '139', '531', '848', '1',  
, '2', '782', '772', '405', '548', '996', '931', '310', '569', '97', '123', '697', '122', '1',  
, '355', '226', '507', '203', '650', '445', '616', '567', '18', '396', '336', '551', '567', '1',
```

Ordenado:

[illegible]

- Tiempo de Ejecución: 4.76837158203125e-06 Segundos

- 3. Método de Ordenamiento por Inserción:** consiste en insertar un elemento del arreglo en la parte izquierda del mismo que ya se encuentra ordenada. Este proceso se repite desde el segundo hasta el (n) enésimo elemento.

Código:

```
import random # Importamos la clase random para los datos randomicos.
import time # Se importa la clase time, para saber el tiempo de ejecucion del programa.

# El siguiente método de ordenamiento se encarga de tomar como parametro una lista.
def ordenamientoPorInsercion(unalista):
    for indice in range(1,len(unalista)): # Este ciclo for, recorre la lista en un rango desde 1 hasta la longitud.
        # La variable "valorActual", sirve como variable temporal, para saber que dato tenemos en esa posicion.
        valorActual = unalista[indice]
        posicion = indice
        # El siguiente ciclo while se utiliza para insertar los valores, con ayuda de un operador lógico, denominado,
        # "and", para que cumpla ambas condiciones y se pueda ordenar la lista.
        while posicion>0 and unalista[posicion-1]>valorActual:
            unalista[posicion]=unalista[posicion-1]
            posicion = posicion-1
        unalista[posicion]=valorActual
#Escribe numeros en un archivo llamado "Ejemplo.txt"

File = open("Ejemplo.txt", "w")
for i in range(10): # Se utiliza el ciclo for, para llenar el archivo con los datos randomicos
    # y separados por comas.
    File.write( str(random.randint(1,1000)) + "," ) # La función Write se utiliza para escribir sobre el archivo,
    # teniendo en cuenta que estan separados por comas.

File.close() # La función Close, se utiliza para cerrar el archivo sobre el cual se esta escribiendo.

# Leer y sumar los numeros del archivo "Ejemplo.txt"
fr = open("Ejemplo.txt", "r")
for line in fr: # El ciclo for en este caso, se utiliza para leer los datos y las líneas del archivo.
    files = line.split(",") # El Método Split se encarga de leer solamente los datos, que no sean comas (,).

tam = len(files)-1 # Len sirve para obtener la longitud.
```

```
# Impresión de los arreglos ordenados y no ordenados.
print ("No Ordenado \n")

print(files)

print ("\nOrdenado \n")

ordenamientoPorInsercion(files)
print(files)

hms = time.time()

# La siguiente linea expresa el tiempo de ejecucion
print((time.time()) - hms)

fr.close()
```



- Ejecución: 100 (cien) Datos

No ordenado:

```
No Ordenado

['938', '582', '458', '429', '925', '187', '25', '798', '330',
'394', '314', '734', '975', '585', '583', '761', '120', '566',
'888', '638', '926', '28', '161', '899', '615', '635', '41', '2
91', '507', '302', '13', '59', '703', '11', '646', '65', '815',
'629', '145', '901', '949', '655', '270', '736', '2', '782', '
625', '640', '84', '616', '5', '195', '494', '857', '330', '189
', '875', '101', '570', '883', '844', '99', '919', '717', '697'
, '52', '459', '57', '162', '138', '563', '626', '39', '484', '
363', '740', '712', '293', '791', '223', '440', '943', '274', '
243', '140', '858', '309', '940', '715', '147', '225', '673', '
135', '628', '84', '723', '43', '127', '591', '691', '']
```

Ordenado:

```
Ordenado

['', '101', '11', '120', '127', '13', '135', '138', '140', '145
', '147', '161', '162', '187', '189', '195', '2', '223', '225',
'243', '25', '270', '274', '28', '291', '293', '302', '309', '
314', '330', '330', '363', '39', '394', '41', '429', '43', '440
', '458', '459', '484', '494', '5', '507', '52', '563', '566',
'57', '570', '582', '583', '585', '59', '591', '615', '616', '6
25', '626', '628', '629', '635', '638', '640', '646', '65', '65
5', '673', '691', '697', '703', '712', '715', '717', '723', '73
4', '736', '740', '761', '782', '791', '798', '815', '84', '84'
, '844', '857', '858', '875', '883', '888', '899', '901', '919'
, '925', '926', '938', '940', '943', '949', '975', '99']
```

- Tiempo de ejecución: 3.0994415283203125e-06 Segundos

- Ejecución: 1000 (Mil) Datos

No ordenado:

```
No Ordenado
```

[	'377'	,	'496'	,	'785'	,	'549'	,	'607'	,	'453'	,	'160'	,	'410'	,	'313'	,	'827'	,	'321'	,	'407'	,	'99'	,	'142'	,	'141'	,	'680'	,	'67'	,	'210'	,	'572'	,	'366'	,	'184'	,	'925'	,	'320'	,	'51'	,	'16'	,	'20'	,	'386'	,	'526'	,	'251'	,	'342'	,	'460'	,	'731'	,	'539'	,	'543'	,	'144'	,	'767'	,	'26'	,	'390'	,	'843'	,	'740'	,	'879'	,	'865'	,	'566'	,	'117'	,	'245'	,	'453'	,	'350'	,	'354'	,	'589'	,	'733'	,	'816'	,	'219'	,	'763'	,	'699'	,	'111'	,	'183'	,	'726'	,	'580'	,	'909'	,	'599'	,	'454'	,	'764'	,	'341'	,	'920'	,	'938'	,	'491'	,	'902'	,	'149'	,	'810'	,	'739'	,	'457'	,	'445'	,	'113'	,	'696'	,	'380'	,	'343'	,	'505'	,	'45'	,	'666'	,	'388'	,	'12'	,	'123'	,	'147'	,	'326'	,	'31'	,	'555'	,	'853'	,	'117'	,	'785'	,	'58'	,	'417'	,	'328'	,	'532'	,	'400'	,	'275'	,	'968'	,	'365'	,	'727'	,	'351'	,	'862'	,	'295'	,	'940'	,	'638'	,	'668'	,	'136'	,	'159'	,	'983'	,	'799'	,	'638'	,	'136'	,	'308'	,	'511'	,	'472'	,	'623'	,	'822'	,	'411'	,	'65'	,	'695'	,	'124'	,	'474'	,	'539'	,	'497'	,	'175'	,	'963'	,	'285'	,	'226'	,	'800'	,	'969'	,	'767'	,	'181'	,	'177'	,	'613'	,	'282'	,	'283'	,	'676'	,	'756'	,	'781'	,	'960'	,	'714'	,	'549'	,	'6'	,	'993'	,	'599'	,	'375'	,	'924'	,	'446'	,	'524'	,	'972'	,	'889'	,	'51'	,	'718'	,	'809'	,	'563'	,	'349'	,	'395'	,	'566'	,	'905'	,	'819'	,	'306'	,	'331'	,	'405'	,	'39'	,	'978'	,	'578'	,	'588'	,	'602'	,	'705'	,	'88'	,	'705'	,	'113'	,	'411'	,	'360'	,	'603'	,	'348'	,	'531'	,	'464'	,	'168'	,	'348'	,	'589'	,	'666'	,	'32'	,	'502'	,	'571'	,	'610'	,	'142'	,	'220'	,	'560'	,	'78'	,	'500'	,	'439'	,	'869'	,	'770'	,	'390'	,	'441'	,	'487'	,	'583'	,	'852'	,	'808'	,	'187'	,	'224'	,	'976'	,	'390'	,	'520'	,	'579'	,	'90'	,	'143'	,	'542'	,	'606'	,	'634'	,	'889'	,	'426'	,	'522'	,	'387'	,	'216'	,	'701'	,	'925'	,	'607'	,	'864'	,	'19'	,	'906'	,	'52'	,	'184'	,	'638'	,	'932'	,	'468'	,	'77'	,	'392'	,	'740'	,	'825'	,	'257'	,	'170'	,	'680'	,	'703'	,	'738'	,	'957'	,	'819'	,	'418'	,	'142'	,	'317'	,	'292'	,	'21'	,	'538'	,	'390'	,	'941'	,	'668'	,	'50'	,	'626'	,	'305'	,	'926'	,	'989'	,	'388'	,	'157'	,	'674'	,	'810'	,	'202'	,	'656'	,	'149'	,	'12'	,	'21'	,	'360'	,	'709'	,	'122'	,	'549'	,	'202'	,	'217'	,	'256'	,	'156'	,	'39'	,	'388'	,	'547'	,	'488'	,	'203'	,	'476'	,	'706'	,	'804'	,	'57'	,	'344'	,	'43'	,	'597'	,	'645'	,	'582'	,	'383'	,	'157'	,	']
---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-----	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---	------	---	-------	---	------	---	-------	---	-------	---	-------	---	-------	---	-------	---	----

Ordenado:

```
Ordenado
```

```
[',', '10', '10', '100', '101', '102', '102', '102', '103', '104',  
'105', '105', '108', '11', '11', '111', '111', '111', '111',  
'113', '113', '115', '115', '117', '117', '117', '118', '118',  
'12', '12', '12', '120', '120', '121', '121', '121', '122', '1'  
22', '123', '124', '124', '125', '126', '127', '127', '127', '1'  
28', '128', '128', '131', '133', '133', '135', '136', '136', '1'  
36', '139', '14', '140', '140', '141', '141', '142', '142', '14'  
2', '142', '143', '143', '144', '144', '145', '145', '146', '14'  
6', '147', '149', '149', '149', '149', '15', '15', '153', '153',  
, '154', '156', '157', '157', '158', '159', '16', '16', '160',  
'160', '162', '164', '165', '166', '167', '167', '168', '168',
```



- Ejecución: 100000 (Cien mil) Datos

No ordenados:

```
765', '885', '543', '930', '720', '866', '600', '266', '735', '
710', '747', '617', '713', '240', '276', '974', '40', '566', '
02', '888', '282', '156', '884', '186', '244', '839', '49', '91
3', '705', '207', '539', '694', '639', '127', '127', '40', '653
', '437', '597', '276', '213', '696', '643', '119', '728', '677
', '899', '481', '84', '555', '41', '659', '23', '392', '39', '
991', '850', '173', '270', '116', '991', '173', '239', '671', '
977', '323', '836', '467', '615', '55', '935', '47', '366', '85
6', '737', '252', '66', '923', '548', '906', '495', '72', '488'
, '240', '896', '67', '403', '448', '241', '180', '633', '367',
, '603', '201', '378', '138', '786', '660', '21', '998', '807',
, '240', '833', '792', '46', '662', '222', '727', '291', '680', '
706', '503', '794', '534', '62', '47', '590', '716', '417', '98
', '959', '555', '986', '569', '16', '784', '217', '751', '13',
, '812', '719', '165', '158', '90', '351', '165', '619', '139',
, '869', '850', '368', '122', '924', '127', '105', '309', '824',
, '774', '355', '824', '83', '258', '78', '939', '855', '945', '9
8', '23', '291', '787', '385', '802', '102', '98', '947', '285'
, '651', '95', '865', '894', '932', '433', '554', '961', '719',
, '687', '295', '85', '390', '211', '398', '108', '327', '751',
, '675', '373', '35', '285', '771', '13', '279', '773', '519', '9
27', '267', '954', '440', '860', '769', '643', '117', '185', '8
06', '938', '978', '535', '805', '527', '141', '820', '39', '97
1', '798', '546', '78', '323', '941', '866', '24', '49', '732',
, '533', '849', '439', '158', '486', '388', '132', '748', '164',
, '79', '788', '497', '683', '773', '276', '368', '487', '448',
, '620', '710', '524', '441', '850', '942', '608', '178', '190',
, '439', '637', '370', '258', '970', '876', '928', '954', '310',
, '31', '114', '856', '328', '660', '621', '563', '168', '393', '
487', '771', '592', '568', '297', '750', '106', '420', '571', '
954', '193', '18', '900', '889', '913', '181', '884', '488', '3
91', '482', '283', '409', '385', '961', '821', '363', '840', '3
60', '459', '611', '75', '522', '195', '485', '367', '878', '84
```

Ordenados:

```
Ordenado
█
```

- Tiempo de ejecución: Indefinido.
- Tiempo de ejecución de datos aleatorios: 3.814697265625e-06 Segundos

- Ejecución: 1000000 (Un millón) de Datos

No ordenados:

```
440', '2776', '2451', '2325', '9737', '6590', '9646', '1658', '1563', '9562', '8863', '4372', '3165', '3598', '8984', '4849', '3824', '3611', '2584', '2217', '5962', '7989', '5748', '4077', '6024', '9037', '4242', '7906', '4527', '5512', '6055', '8425', '7841', '3102', '8887', '3500', '2200', '8488', '4279', '4669', '9948', '6183', '3568', '9259', '9423', '2726', '7479', '5423', '2267', '5451', '7128', '9351', '8499', '1221', '6568', '1605', '1356', '5427', '8557', '9455', '7823', '7094', '4875', '8200', '9096', '7850', '9465', '9466', '3279', '6875', '9796', '2378', '7638', '9458', '2849', '6754', '2657', '4298', '7982', '7687', '8342', '2601', '9404', '4573', '3915', '5978', '9274', '7192', '6596', '5801', '7000', '4928', '5771', '8097', '3049', '5387', '6766', '2109', '9940', '6247', '6280', '1773', '1357', '9260', '1520', '9559', '8810', '7771', '4430', '2367', '9292', '7575', '6345', '4162', '2989', '9958', '2682', '9208', '6952', '7714', '7101', '8799', '9086', '2332', '8743', '1522', '1510', '1320', '1049', '2474', '3598', '8961', '2614', '1487', '4334', '5741', '6632', '3284', '8629', '9035', '1172', '1694', '9856', '2880', '7713', '4764', '1814', '6648', '1667', '3595', '6975', '5252', '8603', '7398', '1126', '1945', '3004', '7102', '8343', '5640', '5704', '3128', '5597', '4521', '5241', '1743', '3650', '5329', '3735', '2508', '9822', '8270', '4015', '1029', '3605', '7983', '5280', '6016', '6176', '3513', '7360', '3927', '4422', '5996', '9250', '4091', '9472', '6363', '2139', '1235', '7469', '3508', '2085', '7027', '9206', '3185', '7039', '1975', '1914', '8365', '6909', '2864', '9680', '2627', '7287', '1521', '5940', '9357', '4896', '4488', '6604', '9359', '1833', '3142', '9023', '3774', '9935', '1334', '7500', '4945', '2509', '3571', '3490', '1151', '8016', '7641', '8948', '6002', '7727', '2290', '8014', '7780', '9679', '4111', '6148', '5934', '3810', '1218', '1564', '6356', '6312', '2747', '7592', '4600', '6780', '2435', '8756', '1779', '6714', '9970', '4046', '4212', '9707', '9691', '4254', '6583', '8264', '8489', '8026', '9826', '
```

Ordenados:

```
Ordenado
█
```

- Tiempo de ejecución: Indefinido
- Tiempo de ejecución de datos aleatorios: 6.4373016357421875e-06 Segundos

## **Conclusiones:**

1. **Método de ordenamiento de burbuja:** Se puede concluir que este método de ordenamiento esta enfocado a su utilización en una estructura de datos de tipo *Array*, siempre y cuando se tenga en cuenta que el numero de datos almacenados en este no debe ser muy grande, además de tener en cuenta su tardío tiempo de ejecución debido a las comparaciones que debe realizar.
2. **Método de ordenamiento QuickSort:** Se concluye que el método QuickSort, como su nombre lo indica es un método de ordenamiento rápido, gracias a su implementación de pivotes y de rápidas comparaciones entre un número determinado de datos incluso de grandes cantidades, es eficaz, en la implementación de estructuras de datos, siempre y cuando se tenga en cuenta que estas deben ser estructuras de datos lineales (*Listas, Pilas, Colas, Arreglos*).
3. **Método de ordenamiento por inserción:** Finalmente el método de ordenamiento por inserción, se pude concluir que a diferencia del método burbuja, su eficiencia, se destaca en la implementación de estructuras de tipo líneas tales como (*Listas, Pilas y Colas*), debido a que su desempeño con un Arreglo no es muy bueno, sobre todo teniendo en cuenta la cantidad de datos que esta estructura posea.