ESTRUCTURA DE DATOS

|  |  |  |  |
| --- | --- | --- | --- |
| Estudiante: | Juan Pablo Silvestre | | |
| Práctica: | Ordenamiento burbuja | Fecha: | No se |

**Burbuja**

|  |  |
| --- | --- |
| Código | Análisis |
| def burbuja(lista): | Define la función llamada burbuja, y se le pasa el parámetro (lista) |
| for i in range(0, len(lista)-1): | Este es el for padre, comienza en 0, y termina dependiendo del tamaño de la lista -1 |
| for j in range(0, len(lista)-i-1): | Este es el for hijo, comienza en 0 y termina dependiendo del tamaño de la lista en base a i y le resta 1. |
| if(lista[j] > lista[j+1]): | Compara si el elemento de la lista que está en la posición j es mayor al elemento que está en la posición siguiente. |
| lista[j], lista[j+1] = lista[j+1], lista[j] | Si el if es verdadero, swapea estos 2 elementos. |
| print(lista) | Imprime la lista ya con la función aplicada. |
| listaimpares = [51, 7, 33, 13, 79, 21] | Se inicia la variable de tipo lista, y tiene 6 elementos. |
| burbuja(listaimpares) | Llama a la función y le pasa como parámetro la lista. |
| Valores   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | # | 51 | 7 | 33 | 13 | 79 | 21 | | 1 | 7 | 51 | 33 | 13 | 79 | 21 | | 2 | 7 | 33 | 51 | 13 | 79 | 21 | | 3 | 7 | 33 | 13 | 51 | 79 | 21 | | 4 | 7 | 33 | 13 | 51 | 21 | 79 | | 5 | 7 | 13 | 33 | 51 | 21 | 79 | | 6 | 7 | 13 | 33 | 21 | 51 | 79 | | 7 | 7 | 13 | 21 | 33 | 51 | 79- | | Ejecución del código (especifique cada una de las variables)  ================================  1# Lista= [51, 7, 33, 13, 79, 21]  i = 0; 5 | j = 0; 5  51 > 7 True  New List = [7, 51, 33, 13, 79, 21]  ================================  2# Lista= [7, 51, 33, 13, 79, 21]  i = 0; 5 | j = 1; 5  51 > 33 True  New List = [7, 33, 51, 13, 79, 21]  ================================  3# Lista= [7, 33, 51, 13, 79, 21]  i = 0; 5 | j = 2; 5  51 > 33 True  v = [7, 33, 13, 51, 79, 21]  ================================  4# Lista= [7, 33, 51, 13, 79, 21]  i = 0; 5 | j = 3; 5  51 > 79 False  New List = [7, 33, 13, 51, 79, 21]  ================================  5# Lista= [7, 33, 51, 13, 79, 21]  i = 0; 5 | j = 4; 5  79 > 21 True  New List = [7, 33, 13, 51, 21, 79]  ================================  ( )# Nada que comparar en j = 5.  ================================  6# Lista= [7, 33, 13, 51, 21, 79]  i = 1; 5 | j = 0; 4  7 > 33 False  New List = [7, 33, 13, 51, 21, 79]  ================================  7# Lista= [7, 33, 13, 51, 21, 79]  i = 1; 5 | j = 1; 4  33 > 13 True  New List = [7, 13, 33, 51, 21, 79]  ================================  8# Lista= [7, 13, 33, 51, 21, 79]  i = 1; 5 | j = 2; 4  33 > 51 False  New List = [7, 13, 33, 51, 21, 79]  ================================  9# Lista= [7, 13, 33, 51, 21, 79]  i = 1; 5 | j = 3; 4  51 > 21 True  New List = [7, 13, 33, 21, 51, 79]  ================================  9# Lista= [7, 13, 33, 21, 51, 79]  i = 1; 5 | j = 4; 4  51 > 79 False  New List = [7, 13, 33, 21, 51, 79]  ================================  10# Lista= [7, 13, 33, 21, 51, 79]  i = 2; 5 | j = 0; 3  7 > 13 False  New List = [7, 13, 33, 21, 51, 79]  ================================  11# Lista= [7, 13, 33, 21, 51, 79]  i = 2; 5 | j = 1; 3  13 > 33 False  New List = [7, 13, 33, 21, 51, 79]  ================================  12# Lista= [7, 13, 33, 21, 51, 79]  i = 2; 5 | j = 2; 3  33 > 21 True  New List = [7, 13, 21, 33, 51, 79]  El loop sigue hasta que i = 5, pero ya no hay nada que comparar. |

**Burbuja mejorado**

|  |  |
| --- | --- |
| Código | Análisis |
| def burbuja\_mejorado(lista): | Define la función llamada burbuja\_mejorado , y se le pasa el parámetro (lista) |
| i=0 | Se inicializa la variable int i con el valor 0 |
| control=True | Se inicia la variable boolean control con el valor True |
| while (i<=len(lista)-2) and control: | Se crea el ciclo while, donde tiene 2 condiciones, la primera es si i es menor al tamaño de la lista menos 2, y la segunda, es si control es true, y está condicionado con el operador lógico and, lo cual significa que el loop seguirá siempre y cuando ambas se cumplan. |
| control=False | Aquí la variable control se le asigna el valor de false. |
| for j in range(0, len(lista)-i-1): | Se crea el for loop, que comienza en 0, y termina en el tamaño de la lista -i-1. |
| if(lista[j] > lista[j+1]): | Compara, si el elemento de la posición j es mayor al de la posición siguiente (j + 1) |
| lista[j], lista[j+1] = lista[j+1], lista[j] | Si el if es verdadero, swapea estos 2 elementos. |
| control=True | Se le asigna a control el valor de true. |
| print(lista) | Se imprime la nueva lista |
| i +=1 | A la variable i se le suma 1. |
| listaimpares = [13, 3, 23, 43, 93, 73] | Se inicializa la variable listaimpares, la cual es una lista de enteros impares. |
| burbuja\_mejorado(listaimpares) | Se llama la función burbuja\_mejorado y se le pasa como parámetro la lista (listaimpares) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Valores   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | # | 13 | 3 | 23 | 43 | 93 | 73 | | 1 | 3 | 13 | 23 | 43 | 93 | 73 | | 2 | 3 | 13 | 23 | 43 | 73 | 93 | | Ejecución del código (especifique cada una de las variables)  ====================  1#  List = [13, 3, 23, 43, 93, 73]  i = 0; 0 <= 4; True | Control = True  j = 0; 5  13 > 13 True  New List = [3, 13, 23, 43, 93, 73]  control=True  ====================  2#  List = [3, 13, 23, 43, 93, 73]  i = 0; 0 <= 4; True | Control = True  j = 1; 5  13 > 23 False  New List = [3, 13, 23, 43, 93, 73]  control=False  ====================  3#  List = [3, 13, 23, 43, 93, 73]  i = 0; 0 <= 4; True | Control = True  j = 2; 5  23 > 43 False  New List = [3, 13, 23, 43, 93, 73]  control=False  ====================  4#  List = [3, 13, 23, 43, 93, 73]  i = 0; 0 <= 4; True | Control = True  j = 3; 5  43 > 93 False  New List = [3, 13, 23, 43, 93, 73]  control=False  ====================  5#  List = [3, 13, 23, 43, 93, 73]  i = 0; 0 <= 4; True | Control = True  j = 4; 5  93 > 73 True  New List = [3, 13, 23, 43, 73, 93]  control=True  ====================  5#  List = [3, 13, 23, 43, 73, 93]  i = 0; 0 <= 4; True | Control = True  j = 5; 5  Nada que comparar.  ===============  6#  List = [3, 13, 23, 43, 73, 93]  i = 1; 0 <= 4; True | Control = True  j = 0; 4  3 > 13 False  New List = [3, 13, 23, 43, 73, 93]  control=False  Note: El ciclo for seguirá, pero la última iteración terminará en un control = false, debido a que ya todo está sorteado, dando por terminado el ciclo while. |

**Burbuja bidireccional**

|  |  |
| --- | --- |
| Código | Análisis |
| def burbuja\_bidir(lista): | Se inicializa la función llamada burbuja\_didir con el parámetro  lista |
| izquierda = 0 | Se inicializa la variable izquierda con el valor 0. |
| derecha = len(lista)-1 | Se inicializa la variable derecha con el valor de la lista - 1 |
| control = True | Se inicializa la variable control con el valor de True |
| while (izquierda < derecha) and control: | Se crea el ciclo while con 2 condiciones, la primera es si izquierda es menor a derecha, y la segunda es si control es igual a true. Y ambas se tienen que cumplir para que el while siga (por el operador lógico and). |
| control = False | Se le asigna a control un nuevo valor (False) |
| for i in range(izquierda, derecha): | Se crea ciclo for donde comienza con el valor que contenga la variable “izquierda” y termina con el valor que contenga la variable “derecha”. |
| if(lista[i] > lista[i+1]): | Compara, si el elemento de la posición i es mayor al de la posición siguiente (i + 1) |
| control = True | Si el if es verdadero, Se le asigna a control el valor de true. |
| lista[i], lista[i+1] = lista[i+1], lista[i] | Si el if es verdadero, swapea estos 2 elementos. |
| print(lista) | Se imprime la nueva lista |
| derecha -= 1 | A la variable derecha se le resta 1. |
| for j in range(derecha, izquierda, -1): | Se crea ciclo for donde comienza con el valor que contenga la variable “izquierda” y termina con el valor que contenga la variable “derecha”. (y se le va r |
| if(lista[j] < lista[j-1]): | Compara, si el elemento de la posición j es menor al de la posición anterior (j - 1) |
| control = True | Se le asigna a control el valor de true. |
| lista[j], lista[j-1] = lista[j-1], lista[j] | Si el if es verdadero, swapea estos 2 elementos. (con el anterior) |
| print(lista) | Se imprime la nueva lista |
| izquierda += 1 | Se le suma a la variable izquierda 1 |
| listaimpares = [31, 33, 5, 87, 19, 49] | Se inicializa la variable listaimpares, la cual es una lista de enteros impares. |
| burbuja\_bidir(listaimpares) | Se llama la función burbuja\_didir y se le pasa como parámetro la lista (listaimpares) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Valores   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | # | 31 | 33 | 5 | 87 | 19 | 49 | | 1 | 31 | 5 | 33 | 87 | 19 | 49 | | 2 | 31 | 5 | 33 | 19 | 87 | 49 | | 3 | 31 | 5 | 33 | 19 | 49 | 87 | | 4 | 31 | 5 | 19 | 33 | 49 | 87 | | 5 | 5 | 31 | 19 | 33 | 49 | 87 | | 6 | 5 | 19 | 31 | 33 | 49 | 87 | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | | Ejecución del código (especifique cada una de las variables)  #1  List = [31, 33, 5, 87, 19, 49]  Izquierda = 0 | derecha = 5 | Control = True  0 < 5 y True  i = 0; 5  31 > 33 False  New list = [31, 33, 5, 87, 19, 49]  =====================  List = [31, 33, 5, 87, 19, 49]  #2 Izquierda = 0 | derecha = 5 | Control = False  0 < 5 y True  i = 1; 5  33 > 5 True  New list = [31, 5, 33, 87, 19, 49]  =====================  List = [31, 5, 33, 87, 19, 49]  #3 Izquierda = 0 | derecha = 5 | Control = True  0 < 5 y True  i = 2; 5  33 > 87 False  New list = [31, 5, 33, 87, 19, 49]  =====================  List = [31, 5, 33, 87, 19, 49]  #4 Izquierda = 0 | derecha = 5 | Control = False  0 < 5 y True  i = 3; 5  87 > 19 True  New list = [31, 5, 33, 19, 87, 49]  =====================  List = [31, 5, 33, 19, 87, 49]]  #5 Izquierda = 0 | derecha = 5 | Control = True  0 < 5 y True  i = 4; 5  87 > 49 True  New list = [31, 5, 33, 19, 49, 87]  ===================================  i = 5; no compara nada.  ====================================  List = [31, 5, 33, 19, 87, 49]]  #6 Izquierda = 0 | derecha = 4 | Control = True  0 < 4 y True  j = 4; 0  49 < 19 False  New list = [31, 5, 33, 19, 49, 87]  =======================================  list = [31, 5, 33, 19, 49, 87]  0 < 4 y True  7# Derecha = 4 | Izquierda = 0; Control = False  j = 3; 0  19 < 33 True  New list = [31, 5, 19, 33, 49, 87]  =======================================  list = [31, 5, 19, 33, 49, 87]  0 < 4 y True  8# Derecha = 4 | Izquierda = 0; Control = True  j = 2; 0  19 < 5 False  New list = [31, 5, 19, 33, 49, 87]  =======================================  list = [31, 5, 19, 33, 49, 87]  0 < 4 y True  9# Derecha = 4 | Izquierda = 0; Control = False  j = 1; 0  5 < 31 True  New list = [5, 31, 19, 33, 49, 87]  =======================================  j = 0; nada que comparer  =======================================  list = [5, 31, 19, 33, 49, 87]  1 < 4 y True  10# Derecha = 4 | Izquierda = 1; Control = True  i = 0; 4  5 > 33 False  New list = [5, 31, 19, 33, 49, 87]  =======================================  list = [5, 31, 19, 33, 49, 87]  1 < 4 y True  11# Derecha = 4 | Izquierda = 1; Control = False  i = 1; 4  33 > 19 True  New list = [5, 19, 31, 33, 49, 87]  ======================================  El loop sigue hasta que se acaba, declarando control = False; luego sigue el for del j, declarando control = False, dando por terminado la ejecución del while ya que una condición para que se ejecute es: control = True. |