

DISEÑO DE PRUEBAS

Diseño de pruebas del ordenamiento BubbleSort

Ordenamiento burbuja

Escenarios

Nombre	Clase	Escenario
Setup1	SortAlgorithm	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Setup2	SortAlgorithm	Conjunto B de 10^4 datos aleatorios.
Setup3	SortAlgorithm	Conjunto A de 10^6 datos aleatorios.
Setup4	SortAlgorithm	{5, 9, 8, 7, 6, 10, 11, 12, 13, 19, 20, 21, 22, 23, 24, 13, 14, 16, 15, 18, 17, 25} // número repetido 13
Setup5	SortAlgorithm	C={-3, -9, -7, -8, -1, 0, 1, 5, 8, 3, 9, 10, 11, 14, 15, 13, 12}
Setup6	SortAlgorithm	{}
Setup7	SortAlgorithm	{4}
Setup8	SortAlgorithm	{10,9,8,7,6,5,4,3,2,1}

Objetivo de la prueba	Ordenar los valores del arreglo por defecto			
Clase	método	escenario	entrada	resultado
SortAlgorithm	1. BubbleSort	Setup1	{1,2,3,4,5,6,7,8,9,10}	{1,2,3,4,5,6,7,8,9,10}

Objetivo de la prueba	Ordenar los valores del arreglo pequeño cuando los elementos de estos están desordenados			
Clase	método	escenario	entrada	resultado
SortAlgorithm	2. BubbleSort	setup2	Conjunto B	Conjunto B en orden ascendente

Objetivo de la prueba	Ordenar los valores del arreglo grande cuando los elementos de estos están desordenados			
Clase	método	escenario	entrada	resultado
SortAlgorithm	3.BubbleSort	setup3	Conjunto A	Conjunto A en orden ascendente

Objetivo de la prueba	Ordenar los valores del arreglo pequeño cuando los elementos de estos están ordenados ascendentemente.			
Clase	método	escenario	entrada	resultado
SortAlgorithm	4.BubbleSort	setup2	Conjunto B	Conjunto B en orden descendiente

Objetivo de la prueba	Ordenar los valores del arreglo grande cuando los elementos de estos están ordenados ascendentemente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	5.BubbleSort	setup3	Conjunto A	Conjunto A en orden descendiente

Objetivo de la prueba	Ordenar los valores del arreglo cuando los elementos de este están en orden descendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	6.BubbleSort	Setup1	Arreglo de valores: {1,2,3,4,5,6,7,8,9,10}	Valores ordenados en orden ascendente {10,9,8,7,6,5,4,3,2,1}

Objetivo de la prueba	Ordenar los valores del arreglo mediano en orden descendente cuando estos entran en orden ascendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	7.BubbleSort	Setup2	Conjunto B	Conjunto B en orden descendente

Objetivo de la prueba	Ordenar los valores del arreglo grande en orden descendente cuando estos valores están en orden ascendente.			
Clase	método	escenario	entrada	resultado
SortAlgorithm	8.BubbleSort	Setup3	Conjunto A	Conjunto A en orden descendente

Objetivo de la prueba	Ordenar los valores del arreglo mediano en orden descendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	9.BubbleSort	Setup2	Conjunto B	Conjunto B ordenado en orden descendente

Objetivo de la prueba	Ordenar los valores del arreglo grande en orden descendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	10.BubbleSort	Setup3	Conjunto A	Conjunto A ordenado en orden descendente

Objetivo de la prueba	Ordenar los valores del arreglo cuando en este hay 2 valores repetidos			
Clase	método	escenario	entrada	resultado
SortAlgorithm	11.BubbleSort	Setup 4	Arreglo de valores enteros desordenados con un número repetido	{5,6,7,8,9,10,11,12,13,13,14,15,16,17,18,19,20,21,22,23,24,25}

Objetivo de la prueba	Ordenar los valores del arreglo cuando en este hay enteros negativos			
Clase	método	escenario	entrada	resultado
SortAlgorithm	12.BubbleSort	Setup5	C	{-9, -8, -7, -3, -1, 0, 1, 3, 5, 8, 9, 10, 11, 12, 13, 14, 15}

Objetivo de la prueba	Ordenar los valores del arreglo cuando este está vacío			
Clase	método	escenario	entrada	resultado
SortAlgorithm	13.BubbleSort	Setup6	Arreglo vacío	Un mensaje de error que muestre que indique que el arreglo está vacío, por tanto no puede ordenar elementos en este.

Objetivo de la prueba	Ordenar los valores del arreglo cuando este solo tiene un elemento			
Clase	método	escenario	entrada	resultado
SortAlgorithm	14.BubbleSort	Setup7	Arreglo de un valor entero	{4}

Objetivo de la prueba	Ordenar los valores del arreglo que entra de manera descendente.			
Clase	método	escenario	entrada	resultado
SortAlgorithm	15.BubbleSort	Setup8	{10,9,8,7,6,5,4,3,2,1}	{1,2,3,4,5,6,7,8,9,10}

Diseño de pruebas del ordenamiento Insertion

Objetivo de la prueba	Ordenar los valores del arreglo por defecto			
Clase	método	escenario	entrada	resultado
SortAlgorithm	1. InsertionSort	Setup1	{1,3,4,5,6,7,2,8,9,10}	{1,2,3,4,5,6,7,8,9,10} Ordenamiento por inserción.

Objetivo de la prueba	Ordenar los valores del arreglo pequeño cuando los elementos de estos están desordenados			
Clase	método	escenario	entrada	resultado
SortAlgorithm	2. InsertionSort	setup2	Conjunto B	Conjunto B en orden ascendente por inserción.

Objetivo de la prueba	Ordenar los valores del arreglo grande cuando los elementos de estos están desordenados			
Clase	método	escenario	entrada	resultado
SortAlgorithm	3.InsertionSort	setup3	Conjunto A	Conjunto A en orden ascendente por inserción.

Objetivo de la prueba	Ordenar los valores del arreglo pequeño cuando los elementos de estos están ordenados ascendentemente.			
Clase	método	escenario	entrada	resultado
SortAlgorithm	4.InsertionSort	setup2	Conjunto B	Conjunto B en orden descendiente.

Objetivo de la prueba	Ordenar los valores del arreglo grande cuando los elementos de estos están ordenados ascendentemente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	5.InsertionSort	setup3	Conjunto A	Conjunto A en orden descendiente.

Objetivo de la prueba	Ordenar los valores del arreglo cuando los elementos de este están en orden descendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	6.InsertionSort	Setup1	Arreglo de valores: {1,3,4,5,6,7,2,8,9,10}	Valores ordenados en orden ascendente {10,9,8,7,6,5,4,3,2,1}

Objetivo de la prueba	Ordenar los valores del arreglo pequeño en orden descendente cuando estos entran en orden ascendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	7.InsertionSort	Setup2	Conjunto A	Conjunto A en orden ascendente.

Objetivo de la prueba	Ordenar los valores del arreglo grande en orden ascendente cuando estos valores están en orden aleatorio.			
Clase	método	escenario	entrada	resultado
SortAlgorithm	8.InsertionSort	Setup3	Conjunto A	Conjunto A en orden ascendente.

Objetivo de la prueba	Ordenar los valores del arreglo pequeño en orden descendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	9.InsertionSort	Setup2	Conjunto B	Conjunto B ordenado en orden descendente.

Objetivo de la prueba	Ordenar los valores del arreglo grande en orden descendente			
Clase	método	escenario	entrada	resultado
SortAlgorithm	10.InsertionSort	Setup3	Conjunto A	Conjunto A ordenado en orden descendente.

Objetivo de la prueba	Ordenar los valores del arreglo cuando en este hay 2 valores repetidos			
Clase	método	escenario	entrada	resultado
SortAlgorithm	11.InsertionSort	Setup 4	Arreglo de valores enteros desordenados con un número repetido	{5,6,7,8,9,10,11,12,13,13,14,15,16,17,18,19,20,21,22,23,24,25}

Objetivo de la prueba	Ordenar los valores del arreglo cuando en este hay enteros negativos			
Clase	método	escenario	entrada	resultado
SortAlgorithm	12.InsertionSort	Setup5	Arreglo de valores enteros desordenados	{-9, -8, -7, -3, -1, 0, 1, 3, 5, 8, 9, 10, 11, 12, 13, 14, 15}

Objetivo de la prueba	Ordenar los valores del arreglo cuando este está vacío			
Clase	método	escenario	entrada	resultado
SortAlgorithm	13.InsertionSort	Setup6	Arreglo vacío	Un mensaje de error que muestre que indique que el arreglo está vacío, por tanto no puede ordenar elementos en este.

Objetivo de la prueba	Ordenar los valores del arreglo cuando este solo tiene un elemento			
Clase	método	escenario	entrada	resultado
SortAlgorithm	14.InsertionSort	Setup7	Arreglo de un valor entero	{4}

Objetivo de la prueba	Ordenar los valores del arreglo que entra de manera descendente.			
Clase	método	escenario	entrada	resultado
SortAlgorithm	15.InsertionSort	Setup8	{10,9,8,7,6,5,4,3,2,1}	{1,2,3,4,5,6,7,8,9,10}

