

Diseño de Pruebas Unitarias

Clase	Método	Escenario	Valores entrada	Resultado
SortAlgorithms	BubbleSort();	El escenario inicializa un arreglo de enteros consecutivos de 10 posiciones ordenado de menor a mayor con los mismos números del arreglo que entra por parámetro	Un arreglo de enteros consecutivos de 10 posiciones ordenado de mayor a menor	Es verdadero puesto que ambos arreglos están ordenados de menor a mayor y contienen los mismos números
SortAlgorithms	BubbleSort();	El escenario inicializa un arreglo de enteros positivos aleatorios de 10 posiciones	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$
SortAlgorithms	BubbleSort();	El escenario inicializa un arreglo de enteros consecutivos de 10000 posiciones ordenado de menor a mayor con los mismos números del arreglo que entra por parámetro	Un arreglo de enteros consecutivos de 10000 posiciones ordenado de mayor a menor	Es verdadero puesto que ambos arreglos están ordenados de menor a mayor y contienen los mismos números
SortAlgorithms	BubbleSort();	El escenario inicializa un arreglo de enteros positivos aleatorios de	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$

		10000 posiciones		
SortAlgorithms	BubbleSort();	El escenario inicializa un arreglo de enteros consecutivos de 10 posiciones ordenado de mayor a menor donde hay un par de valores repetidos	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$
SortAlgorithms	BubbleSort();	El escenario inicializa un arreglo de enteros consecutivos de 10000 posiciones ordenado de mayor a menor donde hay un par de valores repetidos	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$

Objetivo: probar si el método de BubbleSort ordena cualquier arreglo de enteros de menor a mayor.

Clase	Método	Escenario	Valores entrada	Resultado
SortAlgorithms	InsertionSort();	El escenario inicializa un arreglo de enteros consecutivos de 10 posiciones ordenado de menor a mayor con los mismos números del arreglo que entra por parámetro	Un arreglo de enteros consecutivos de 10 posiciones ordenado de mayor a menor	Es verdadero puesto que ambos arreglos están ordenados de menor a mayor y contienen los mismos números
SortAlgorithms	InsertionSort ();	El escenario inicializa un arreglo de enteros positivos aleatorios de 10 posiciones	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$
SortAlgorithms	InsertionSort ();	El escenario inicializa un arreglo de enteros consecutivos de 10000 posiciones ordenado de menor a mayor con los mismos números del arreglo que entra por parámetro	Un arreglo de enteros consecutivos de 10000 posiciones ordenado de mayor a menor	Es verdadero puesto que ambos arreglos están ordenados de menor a mayor y contienen los mismos números
SortAlgorithms	InsertionSort ();	El escenario inicializa un arreglo de enteros positivos aleatorios de 10000 posiciones	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$

SortAlgorithms	InsertionSort ();	El escenario inicializa un arreglo de enteros consecutivos de 10 posiciones ordenado de mayor a menor donde hay un par de valores repetidos	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$
SortAlgorithms	InsertionSort ();	El escenario inicializa un arreglo de enteros consecutivos de 10000 posiciones ordenado de mayor a menor donde hay un par de valores repetidos	El arreglo inicializado en el escenario	Es verdadero puesto que para cualquier posición i del arreglo, se cumple que $i \geq i+1$

Objetivo: probar si el método de InsertionSort ordena cualquier arreglo de enteros de menor a mayor.