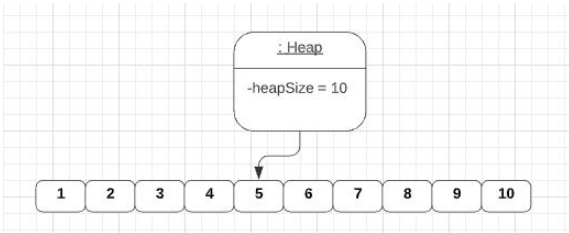
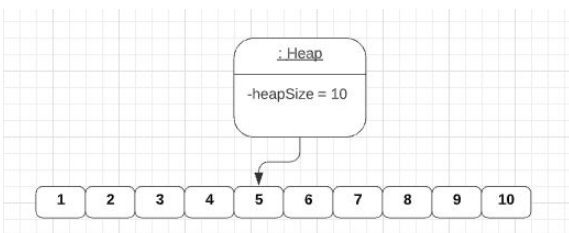
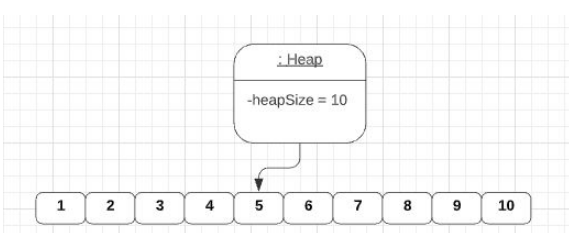
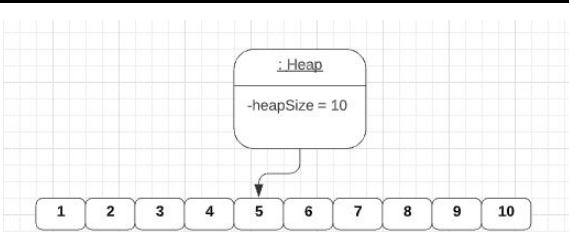
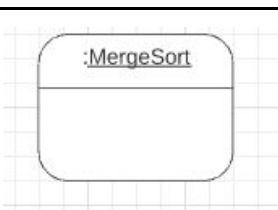
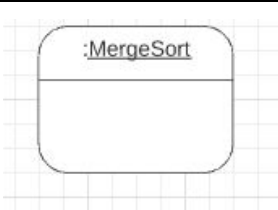
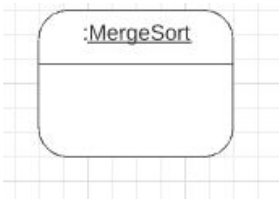
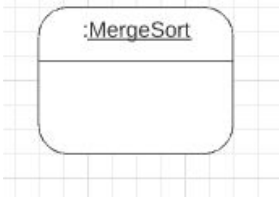
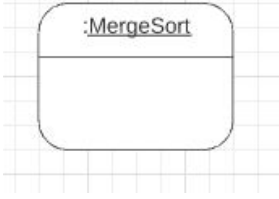
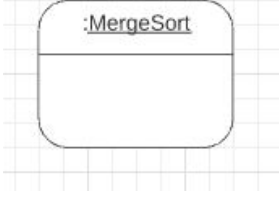
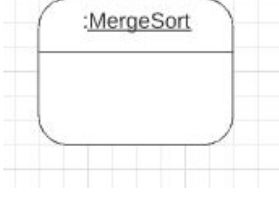
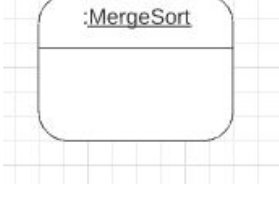
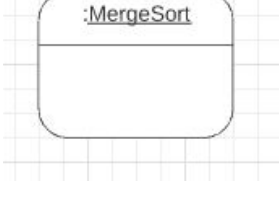


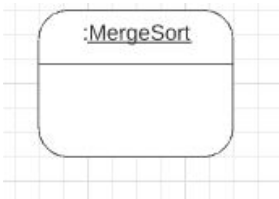
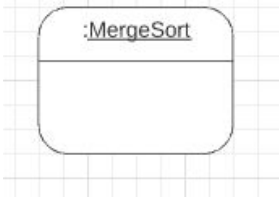
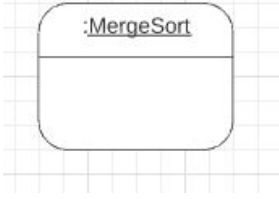
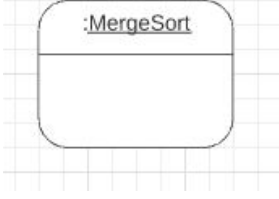
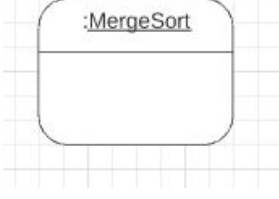
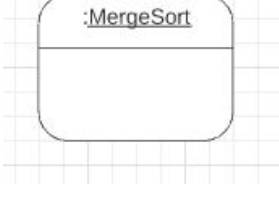
1. Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenary1()	HeapSortTest	
setupScenary2()	HeapSortTest	
setupScenary3()	HeapSortTest	
setupScenary4()	HeapSortTest	
setupScenary5()	HeapSortTest	

setupScenary6()	HeapSortTest	
setupScenary7()	HeapSortTest	
setupScenary8()	HeapSortTest	
setupScenary9()	HeapSortTest	
setupScenary10()	HeapSortTest	
setupScenary11()	HeapSortTest	

setupScenary12()	HeapSortTest	 <p>The diagram shows a class <code>:Heap</code> with an attribute <code>-heapSize = 10</code>. An arrow points from the <code>Heap</code> object to an array of 10 elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.</p>
setupScenary13()	HeapSortTest	 <p>The diagram shows a class <code>:Heap</code> with an attribute <code>-heapSize = 10</code>. An arrow points from the <code>Heap</code> object to an array of 10 elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.</p>
setupScenary14()	HeapSortTest	 <p>The diagram shows a class <code>:Heap</code> with an attribute <code>-heapSize = 10</code>. An arrow points from the <code>Heap</code> object to an array of 10 elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.</p>
setupScenary15()	HeapSortTest	 <p>The diagram shows a class <code>:Heap</code> with an attribute <code>-heapSize = 10</code>. An arrow points from the <code>Heap</code> object to an array of 10 elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.</p>
setupScenary1()	MergeSortTest	 <p>The diagram shows a class <code>:MergeSort</code>.</p>
setupScenary2()	MergeSortTest	 <p>The diagram shows a class <code>:MergeSort</code>.</p>

setupScenary3()	MergeSortTest	
setupScenary4()	MergeSortTest	
setupScenary5()	MergeSortTest	
setupScenary6()	MergeSortTest	
setupScenary7()	MergeSortTest	
setupScenary8()	MergeSortTest	
setupScenary9()	MergeSortTest	

setupScenary10()	MergeSortTest	
setupScenary11()	MergeSortTest	
setupScenary12()	MergeSortTest	
setupScenary13()	MergeSortTest	
setupScenary14()	MergeSortTest	
setupScenary15()	MergeSortTest	

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos generados aleatoriamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	TenElementsRandomOrderTest	setupScenariy1	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados ascendentemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	TenElementsAscendingOrderTest	setupScenariy2	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados descendentemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	TenElementsDescendingOrderTest	setupScenariy3	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera aleatoria.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	HundredElementsRandomOrderTest	setupScenario4	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados ascendentemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	HundredElementsAscendingOrderTest	setupScenario5	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados descendientemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	HundredElementsDescendingOrderTest	setupScenario6	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera aleatoria.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	ThousandElementsRandomOrderTest	setupScenariy7	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	ThousandElementsAscendingOrderTest	setupScenariy8	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera descendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	ThousandElementsDescendingOrderTest	setupScenariy9	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera aleatoria..

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	TenThousandElementsRandomOrderTest	setupScenario10	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	TenThousandElementsAscendingOrderTest	setupScenario11	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados descendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	TenThousandElementsDescendingOrderTest	setupScenario12	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera aleatorio.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	HundredThousandElementsRandomOrderTest	setupScenario13	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	HundredThousandElementsAscendingOrderTest	setupScenario14	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo heap sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
HeapSortTest	HundredThousandElementsDescendingOrderTest	setupScenario15	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos generados aleatoriamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	TenElementsRandomOrderTest	setupScenariy1	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados ascendentemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	TenElementsAscendingOrderTest	setupScenariy2	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados descendientemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	TenElementsDescendingOrderTest	setupScenariy3	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera aleatoria.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	HundredElementsRandomOrderTest	setupScenariy4	int[] newArray = heap.GetTree()	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados ascendentemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	HundredElementsAscendingOrderTest	setupScenariy5	int[] newArray = heap.GetTree()	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados descendientemente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	HundredElementsDescendingOrderTest	setupScenariy6	int[] newArray = heap.GetTree()	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera aleatoria.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	ThousandElementsRandomOrderTest	setupScenariy7	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	ThousandElementsAscendingOrderTest	setupScenariy8	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera descendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	ThousandElementsDescendingOrderTest	setupScenariy9	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera aleatoria..

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	TenThousandElementsRandomOrderTest	setupScenario10	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	TenThousandElementsAscendingOrderTest	setupScenario11	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados descendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	TenThousandElementsDescendingOrderTest	setupScenario12	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera aleatorio.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	HundredThousandElementsRandomOrderTest	setupScenariy13	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	HundredThousandElementsAscendingOrderTest	setupScenariy14	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.

Objetivo de la Prueba: Verificar que el algoritmo merge sort ordene correctamente con elementos inicialmente ordenados de manera ascendente.

Clase	Método	Escenario	Valores de Entrada	Resultado
MergeSortTest	HundredThousandElementsDescendingOrderTest	setupScenariy15	<code>int[] newArray = heap.GetTree()</code>	Se ha ordenado correctamente el conjunto de valores.