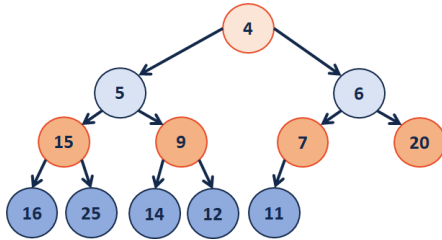


Formally, a binary tree T is a minHeap if:

-
-

Heap Operation: insert / bubbleUp:



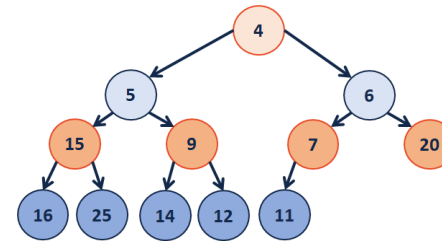
-	4	5	6	15	9	7	20	16	25	14	12	11			
---	---	---	---	----	---	---	----	----	----	----	----	----	--	--	--

-	4	5	6	15	9		20	16	25	14	12	11			
---	---	---	---	----	---	--	----	----	----	----	----	----	--	--	--

-	4	5		15	9		20	16	25	14	12	11			
---	---	---	--	----	---	--	----	----	----	----	----	----	--	--	--

Heap.java (partial)	
1	<code>void insert(int key) {</code>
2	<code> // Check to ensure there's space to insert an element</code>
3	<code> if (size==CAPACITY) { throw new FullHeapException() }</code>
4	
5	<code> // Insert the new element at the end of the array</code>
6	<code> data[++size] = key;</code>
7	
8	<code> bubbleUp(size); // Restore the heap property</code>
9	<code>}</code>
31	<code>void bubbleUp(_____) {</code>
32	<code> if (_____) {</code>
33	<code> if (data[index] < data [_____]) {</code>
34	<code> swap(data, index, index/2);</code>
35	<code> bubbleUp(_____);</code>
36	<code> }</code>
37	<code> }</code>
38	<code>}</code>

Heap Operation: removeMin / bubbleDown



-	4	5	6	15	9	7	20	16	25	14	12	11			
---	---	---	---	----	---	---	----	----	----	----	----	----	--	--	--

-		5	6	15	9	7	20	16	25	14	12				
---	--	---	---	----	---	---	----	----	----	----	----	--	--	--	--

-			6	15	9	7	20	16	25	14	12				
---	--	--	---	----	---	---	----	----	----	----	----	--	--	--	--

-			6	15		7	20	16	25	14	12				
---	--	--	---	----	--	---	----	----	----	----	----	--	--	--	--

Heap.java (partial)	
1	<code>int removeMin() {</code>
2	<code> // Swap with the last value</code>
3	<code> int minValue = data[1];</code>
4	<code> data[1] = data[size];</code>
5	<code> data[size] = minValue;</code>
6	<code> size--;</code>
7	
8	<code> bubbleDown(1); // Restore the heap property</code>
9	<code> return minValue; // Return the minimum value</code>
10	<code>}</code>
1	<code>void bubbleDown(int index) {</code>
2	<code> if (!isLeaf(index)) {</code>
3	<code> int minChildIndex = minChild(index);</code>
4	<code> if (data[index] > data[minChildIndex]) {</code>
5	<code> swap(data, index, minChildIndex);</code>
6	<code> bubbleDown(_____);</code>
7	<code> }</code>
8	<code> }</code>
9	<code>}</code>
10	