Array Representation of Heap (using level-order numbers as indices)

```
  41           A[0]    ---
 /\           /\        
 / \         / \      
 / \        / \    
 / \      / \      
 / \        / \    
2    7
```

**BubbleUp:** (insert)

Given an array, \( H[1..N+1] \) of elements where \( H[1]...H[N] \) form a max heap and \( H[k] \) is the heap element at the \( k \)th level-order location in the tree, "add" element \( H[N+1] \) to the heap to make a max heap of size \( N+1 \).

\[
i = N+1;
\]
\[
\text{while } H[i] < H[i/2] \\
\text{exchange } H[i], H[i/2] \\
i = i/2
\]

"Pictorially"

```
       /
      /\  
Max  
 /\  
 Heap 
 /\  
-------
```

**BubbleDown:**

Given a Tree, \( T \), with root \( r \), in which each of \( r \)'s children are max heaps, convert \( T \) to a max heap (assume \( T \) is stored in same level-order arrangement described above.) \( j \) is defined to be the index of the larger of the two children of \( T[i] \).

\[
i = 1;
\]
\[
j = \text{index of Max}(T[2*i],T[2*i+1])
\]
\[
\text{while } i < N/2 \text{ AND } T[i] < T[j] \\
\text{exchange } T[i], T[j] \\
i = j \\
j = \text{index of Max}(T[2*i],T[2*i+1])
\]

"Pictorially"

```
r
 / \  
H1 H2
```

where \( H1 \) and \( H2 \) are max heaps: