A New Data Structure Arrives:

**ADT:**
- insert
- remove
- isEmpty

### Implementation of _______________

<table>
<thead>
<tr>
<th>insert</th>
<th>removeMin</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>O(n)</td>
<td>O(n)</td>
<td>Unsorted Array</td>
</tr>
<tr>
<td>O(1)</td>
<td>O(n)</td>
<td>Unsorted List</td>
</tr>
<tr>
<td>O(lg(n))</td>
<td>O(1)</td>
<td>Sorted Array</td>
</tr>
<tr>
<td>O(lg(n))</td>
<td>O(1)</td>
<td>Sorted List</td>
</tr>
</tbody>
</table>

Q1: What errors exist in this table? (Fix them!)

Q2: Which algorithm would we use?

---

**Implementing a (min)Heap as an Array**

Operations:
- leftChild(index) :=
- rightChild(index) :=
- parent(index) :=

**Inserting into a Heap**

...running time?
What’s wrong with this code?

Heap Operation: removeMin / heapifyDown:

Q: How do we construct a heap given data?