

1. **6.1-4** Where in a max-heap might the smallest element reside, assuming that all elements are distinct? The smallest element in a max-heap must be a leaf with no children. In a complete tree, it *must* be in the bottom level, as all other levels have children. In a non-complete tree, it will be either in the bottom level or in the second to last level, as only one level is allowed to be incomplete in a heap tree.

Let us consider what would happen if the smallest element had a child. In that case, we would have a subtree where the parent is greater than the child, violating the max-heap property that all subtrees within a max tree must be max trees.

2. **6.1-5** Is an array that is sorted in order a min-heap?

A min-heap is a heap where each parent is smaller than either of its children. An array sorted in increasing order will definitionally have all values to the right of any given element be larger than that element. More formally, this means $\forall j, i : \{j > i \iff a[j] > a[i]\}$, where a is a sorted array.

By the definition of a heap, we have that the children of any element indexed at i is indexed by $2i$ and $2i + 1$. We also have the mathematical property that for any positive number (array indices must be positive integers) $\forall x : 2x > x$. This means that the index of each child *must* be larger than the index of its parent, and consequently must have a larger value. Thus, as all children are larger than the parent we have a min-heap.

3. **6.1-6** Is the array with values 23,17,14,6,13,10,1,5,7,12 a max-heap?

An array is only a max-heap if $\forall i \in a : a[i] > a[2i] \wedge a[i] > a[2i + 1]$. We prove that this property does not hold for this array directly.

Consider $a[4] = 6$. Then $a[8]$ and $a[9]$ *must* be larger than 6, but $a[9] = 7$. Therefore, the max-heap property does not hold.

4. **6.1-7** Show that, with the array representation for storing an n -element heap, the leaves are the nodes indexed by $\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2 \cdots n$.

We proceed by direct proof. First, we show that no node with children can exist in the range $\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2 \cdots n$. Let i be the array index equal to $\lfloor \frac{n}{2} \rfloor + 1$, the lower bound of the range given. Then the children of i must be at $2i$ and $2i + 1$. Through substitution, we have the children at $2\lfloor \frac{n}{2} \rfloor + 2$ and $2\lfloor \frac{n}{2} \rfloor + 3$. These indices are both clearly outside the range n , as $\forall n : 2\lfloor \frac{n}{2} \rfloor + 2 > n$. As this case is the lower bound of the range, then all other indices of the range must also be out of bounds.

We have confirmed that all elements in that range are leafs, but let us show that *all* leaves must be in this range. Consider a node i with no children. Then $2i > n \wedge 2i + 1 > n$. Consequently, it must be that $i > \frac{n}{2}$. Thus, the lowest bound for a leaf is $\frac{n}{2} + 1$, which is precisely the range we are given.