MTech CSE – 1st Semester
Student ID: **A125023**
Student Name: **SRIJITA VERMA**

Question

In an array of size n representing a binary heap, prove that all leaf nodes are located at indices from $\left\lfloor \frac{n}{2} \right\rfloor + 1$ to n.

Answer

Background: Array Representation of a Binary Heap

A binary heap is a complete binary tree stored in an array using level-order indexing. For an array indexed from 1 (standard heap convention):

- The parent of the node at index i is at index $\left\lfloor \frac{i}{2} \right\rfloor$.
- The left child of the node at index i is at index $2i$.
- The right child of the node at index i is at index $2i + 1$.

A leaf node is defined as a node that has no children.

Key Observation

A node at index i has at least one child if and only if:

$$2i \leq n$$

This is because the left child index is $2i$. If $2i > n$, then both left and right child indices exceed the array size.

Thus:

- Nodes with $2i \leq n$ are internal nodes.
- Nodes with $2i > n$ are leaf nodes.

Step-by-Step Proof

Step 1: Identify the Largest Index of an Internal Node

We want the largest index i such that:

$$2i \leq n$$

Solving:

$$i \leq \frac{n}{2}$$

Since array indices are integers:

$$i \leq \left\lfloor \frac{n}{2} \right\rfloor$$

Thus, all indices from 1 to $\left\lfloor \frac{n}{2} \right\rfloor$ correspond to internal nodes.

Step 2: Identify Leaf Node Indices

Any index i such that:

$$i > \left\lfloor \frac{n}{2} \right\rfloor$$

will satisfy:

$$2i > n$$

Hence, these nodes cannot have children and are therefore leaf nodes.

Step 3: Determine the Range of Leaf Nodes

From the above observations:

- Internal nodes are at indices 1 to $\left\lfloor \frac{n}{2} \right\rfloor$.
- Leaf nodes are at indices $\left\lfloor \frac{n}{2} \right\rfloor + 1$ to n.

ASCII Tree Diagram for a Binary Heap

Consider a binary heap with $n = 10$ elements.

Tree Representation

```
        1
      /   \
     2     3
    / \   / \
   4   5 6   7
  / \
 8   9
/
10
```

Corresponding Array Indices

Index	1	2	3	4	5	6	7	8	9	10
Node	R	L1	L1	L2	L2	L2	L2	L3	L3	L3

Explanation Using the Diagram

Nodes at indices 1 to 5 have at least one child.

For example, node 5 has a left child at index:

$$2 \times 5 = 10$$

Nodes at indices 6 to 10 have no children. For any index $i \geq 6$:

$$2i > 10$$

and therefore no child exists.

Since:

$$\left\lfloor \frac{10}{2} \right\rfloor = 5,$$

we conclude:

- Indices 1 to 5 correspond to internal nodes.
- Indices 6 to 10 correspond to leaf nodes.

This visually confirms that all leaf nodes occupy indices:

$$\left\lfloor \frac{n}{2} \right\rfloor + 1 \text{ to } n.$$

Diagram Explanation (In Words)

The array representation of a binary heap can be visualized as a complete binary tree arranged level by level from left to right. The root of the heap corresponds to index 1 in the array. The next level of the tree corresponds to indices 2 and 3, followed by indices $4, 5, 6$, and 7 at the next level, and so on. Each level of the tree is filled completely before nodes are added to the next level.

In this representation, each node at index i has its left child at index $2i$ and its right child at index $2i + 1$, provided these indices do not exceed n, the size of the array. Visually, this means that nodes in the upper levels of the tree always have children, while nodes near the bottom may not.

The last level of the tree is the only level that may be partially filled. All nodes at this level appear consecutively at the end of the array. These nodes do not have children because there are no remaining positions in the array to store them. As a result, they are leaf nodes.

If we observe the array indices, the transition from internal nodes to leaf nodes occurs exactly after index $\left\lfloor \frac{n}{2} \right\rfloor$. Nodes at indices 1 through $\left\lfloor \frac{n}{2} \right\rfloor$ have at least one child and therefore correspond to internal nodes in the tree. Nodes at indices $\left\lfloor \frac{n}{2} \right\rfloor + 1$ through n appear in the lowest level of the tree and have no children, which visually confirms that they are leaf nodes.

Thus, when the array is interpreted as a level-order traversal of a complete binary tree, it becomes clear that all leaf nodes are grouped at the end of the array, occupying indices $\left\lfloor \frac{n}{2} \right\rfloor + 1$ through n.

Illustrative Example

Let $n = 10$.

$$\left\lfloor \frac{10}{2} \right\rfloor = 5$$

- Indices 1 to 5 are internal nodes.
- Indices 6 to 10 are leaf nodes.

All nodes from index 6 onward have no children, confirming the result.

Why This Holds for All Binary Heaps

This property holds independently of whether the heap is a min-heap or a max-heap, because it depends solely on:

- The complete binary tree structure.
- The array indexing scheme.

It is not affected by key values or heap order.

Final Conclusion

In an array of size n representing a binary heap, all leaf nodes are located at indices:

$$\left\lfloor \frac{n}{2} \right\rfloor + 1 \text{ to } n.$$

This result follows directly from the array representation of a complete binary tree and the definition of a leaf node.