MTech CSE – 1st Semester
Student Name: SIDDHARTH KUMAR
Student ID: A125021

Question 2

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:

Binary Heap Representation

A binary heap is a complete binary tree that is commonly stored using an array representation. The nodes are placed in the array following level-order traversal.

Assuming the array is indexed starting from 1, the relationships between nodes are:

- Parent of node at index i is at $\left\lfloor \frac{i}{2} \right\rfloor$,
- Left child of node at index i is at $2i$,
- Right child of node at index i is at $2i + 1$.

A node is called a *leaf node* if it does not have any children.

Important Observation

A node at index i can have at least one child only if its left child index exists within the array bounds. This condition can be written as:

$$2i \leq n$$

If $2i > n$, then neither the left nor the right child can exist.

Hence:

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

Proof

Step 1: Maximum Index of an Internal Node

For a node to be an internal node, it must satisfy:

$$2i \leq n$$

Solving for i:

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

Since i must be an integer:

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

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

Step 2: Identifying Leaf Nodes

Any index i such that:

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

will automatically satisfy:

$$2i > n$$

Therefore, nodes with indices greater than $\left\lfloor \frac{n}{2} \right\rfloor$ cannot have children and must be leaf nodes.

Step 3: Range of Leaf Nodes

From the above results:

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

Illustrative Example

Let the heap contain $n = 10$ elements.

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

- Indices 1 through 5 represent internal nodes.
- Indices 6 through 10 represent leaf nodes.

Visual Representation

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

In this tree, nodes at indices 6 to 10 do not have any children, confirming that they are leaf nodes.

Why This Property Always Holds

This result depends only on the structural properties of a complete binary tree and the array indexing scheme used to represent it. It is independent of whether the heap is a min-heap or a max-heap, since the ordering of keys does not affect node positions.

Final Conclusion

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

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

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