



MTech CSE – 1st Semester

Student Name: SIDDHARTH KUMAR**Student ID:** A125021

Question 3(a)

Show that in a heap containing n elements, the number of nodes whose height is exactly h is at most:

$$\left\lceil \frac{n}{2^{h+1}} \right\rceil$$

Answer:

Preliminaries

We consider a binary heap represented as a complete binary tree. The ordering property of the heap (min-heap or max-heap) is irrelevant here; only the structural properties of the tree are used in the proof.

Height Definition: The height of a node is the number of edges on the longest downward path from that node to any leaf. Consequently:

- Leaf nodes have height 0,
- Nodes directly above leaves have height 1,
- The root has the maximum height.

Objective

Let:

- n be the total number of nodes in the heap,

- $h \geq 0$ be a fixed height,
- N_h be the number of nodes with height exactly h .

We aim to prove:

$$N_h \leq \left\lceil \frac{n}{2^{h+1}} \right\rceil.$$

Key Idea

A node of height h must have a sufficiently large subtree beneath it. Since the heap is complete, such subtrees cannot overlap and must together fit within the total n nodes of the heap.

Minimum Subtree Size for Height h

Consider any node whose height is h . The subtree rooted at this node must extend downward for h levels.

The minimum number of nodes in a binary tree of height h is obtained when the tree is perfectly balanced:

$$1 + 2 + 4 + \cdots + 2^h = 2^{h+1} - 1.$$

For an upper-bound argument, it is sufficient to use the weaker inequality:

$$\text{Subtree size} \geq 2^{h+1}.$$

This simplification preserves correctness while making the counting argument clearer.

Disjointness of Subtrees

No two nodes of the same height h can lie on the same root-to-leaf path. Hence, no node of height h can be an ancestor of another node of height h .

Therefore, the subtrees rooted at nodes of height h are pairwise disjoint.

Counting Argument

Let N_h denote the number of nodes of height h . Each such node contributes at least 2^{h+1} nodes through its subtree.

Hence, the total number of nodes covered by all such subtrees is at least:

$$N_h \cdot 2^{h+1}.$$

Since the entire heap contains only n nodes:

$$N_h \cdot 2^{h+1} \leq n.$$

Final Inequality

Dividing both sides by 2^{h+1} :

$$N_h \leq \frac{n}{2^{h+1}}.$$

Because N_h must be an integer, we take the ceiling:

$$N_h \leq \left\lceil \frac{n}{2^{h+1}} \right\rceil.$$

Conclusion

In a heap of size n , the number of nodes with height h is bounded above by:

$$\left\lceil \frac{n}{2^{h+1}} \right\rceil$$

This result reflects the structural property of heaps: most nodes are concentrated near the leaves, while very few nodes exist at greater heights. This bound is fundamental in proving that heap construction algorithms such as BUILD-HEAP run in linear time.