A Heap Data Structure
(specifically a minHeap in this example, as the minimum element is at the root)

Given an index \( i \), it’s parent and children can be reached in O(1) time:
- \( \text{leftChild} := 2i \)
- \( \text{rightChild} := 2i + 1 \)
- \( \text{parent} := \text{floor}(i / 2) \)

Formally, a complete binary tree \( T \) is a minHeap if:
- \( T = {} \) or
- \( T = \{r, T_L, T_R\} \) and \( r \) is less than the roots of \( T_L, T_R \) and \( T_L, T_R \) are minHeaps

Inserting into a Heap

How do we complete this code?

Running time of insert?

Heap Operation: removeMin / heapifyDown:
Q: How do we construct a heap given data?

Theorem: The running time of buildHeap on array of size n is: ________.

Strategy:

Define S(h):
Let S(h) denote the sum of the heights of all nodes in a complete tree of height h.

\[ S(0) = \]
\[ S(1) = \]
\[ S(h) = \]

Proof of S(h) by Induction:

Finally, finding the running time:

Running Time?

CS 225 – Things To Be Doing:

1. Theory Exam 3 starts on Thursday (topic list online, more soon!)
2. MP5 due date: Monday, November 5\textsuperscript{th}
3. lab_hash is due Sunday, November 4\textsuperscript{th}
4. Daily POTDs are ongoing!