Assume $n = 2^h - 1$ (h is the height of the tree). The total number of swaps by the MaxHeapBottomUp algorithm is

$$
\begin{aligned}
&\leq \sum_{i=0}^{h-1} i \times 2^{h-1-i} \\
&= 2^{h-1} \times \sum_{i=1}^{h-1} i/2^i \\
&= 2^{h-1} \times (2 - \frac{h+1}{2^{h-1}}) \\
&= 2^h - h - 1 \\
&\in \mathcal{O}(n)
\end{aligned}
$$

Data Structures

2022-11-03

Assume $n = 2^h - 1$ (h is the height of the tree). The total number of swaps by the MaxHeapBottomUp algorithm is

$$
\begin{aligned}
&\leq \sum_{i=0}^{h-1} i \times 2^{h-1-i} \\
&= 2^{h-1} \times \sum_{i=1}^{h-1} i/2^i \\
&= 2^{h-1} \times (2 - \frac{h+1}{2^{h-1}}) \\
&= 2^h - h - 1 \\
&\in \mathcal{O}(n)
\end{aligned}
$$

Alternatively, multiply the summation by two.

$$2\sum_{i=1}^{h-1} i \times 2^{h-1-i} = 1\times 2^{h-1} + 2\times 2^{h-2} + 3\times 2^{h-3} + \cdots + (h-1)\times 2^1$$

Substract the given summation below from the equation above.

$$\sum_{i=1}^{h-1} i \times 2^{h-1-i} = 1\times 2^{h-2} + 2\times 2^{h-3} + 3\times 2^{h-4} + \cdots + (h-1)\times 2^0$$

Then, we obtain

$$
\begin{aligned}
\sum_{i=1}^{h-1} i \times 2^{h-1-i} &= 2^{h-1} + 2^{h-2} + 2^{h-3} + \cdots + 2^1 - (h-1)\times 2^0 \\
&= 2^{h-1} + 2^{h-2} + 2^{h-3} + \cdots + 2^1 + 2^0 - h \\
&= \sum_{i=0}^{h-1} 2^i - h = \frac{1-2^h}{1-2} - h = 2^h - h - 1
\end{aligned}
$$