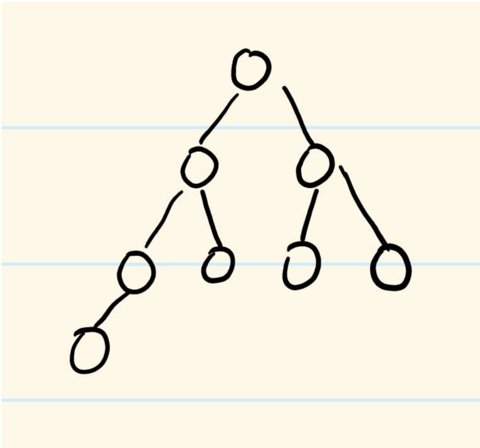
Assignment 1

Qiming Lyu

2025-09-11

1. The smallest binary heap B_h of height h is essentially a perfect binary tree of height $h-1$ plus a leaf node on the most bottom left corner, as shown in the figure below.

   The number of nodes in B_h, denoted as n_h, is given by:

   $$n_h = 2^h$$

   Proof. We know that a perfect binary tree of height $h-1$ has $2^h - 1$ nodes. By adding one more node to the most bottom left corner, we get:

   $$n_h = (2^h - 1) + 1 = 2^h$$

   □

2. The largest binary heap B_h of height h is a perfect binary tree of height h, which has $2^{h+1} - 1$ nodes.

   Therefore, the size n of a binary heap of height h is bounded by:

   $$2^h \leq n \leq 2^{h+1} - 1$$

3. From the inequality from question 2, we have:

   $$2^h \leq n \leq 2^{h+1} - 1 < \mathbf{2^{h+1}}$$

   Taking the base-2 logarithm of all sides, we get:

   $$h \leq \lg n < h + 1$$

Because h is an integer and because of the definition of the floor function, we have:

$$h = \lfloor \lg n \rfloor$$

4. The index of the last internal node in a binary heap of size n is

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

Proof. Suppose the height of the binary heap is h. And we divide the binary heap into two parts: the last level and the rest of the tree. Say there are k $\left(0 < k \leq 2^h\right)$ nodes in the last level, the size of the tree n is given by:

$$n = \left(2^h - 1\right) + k.$$

The index I of the last internal node is the index of the parent of the last node, which is given by:

$$I = \left(2^{h-1} - 1\right) + \left\lceil \frac{k}{2} \right\rceil .$$

Since

$$\begin{aligned} \left\lfloor \frac{n}{2} \right\rfloor &= \left\lfloor \left(2^{h-1} - \frac{1}{2}\right) + \frac{k}{2} \right\rfloor \quad , \\ &= \left\lfloor \left(2^{h-1} - 1\right) + \frac{k}{2} + \frac{1}{2} \right\rfloor \\ &= \left(2^{h-1} - 1\right) + \left\lfloor \frac{k}{2} + \frac{1}{2} \right\rfloor \end{aligned}$$

we only need to prove that

$$\left\lfloor \frac{k}{2} + \frac{1}{2} \right\rfloor = \left\lceil \frac{k}{2} \right\rceil .$$

There are two cases: when k is even and when k is odd, that is,

$$k = 2i \text{ or } k = 2i + 1, \quad i \in \mathbb{Z}.$$

Case 1 $k = 2i$:

$$\text{LHS} = \left\lfloor \frac{k}{2} + \frac{1}{2} \right\rfloor = \left\lfloor i + \frac{1}{2} \right\rfloor = i,$$

$$\text{RHS} = \left\lceil \frac{k}{2} \right\rceil = \lceil i \rceil = i.$$

Thus, LHS $=$ RHS.

Case 2 $k = 2i + 1$:

$$\text{LHS} = \left\lfloor \frac{k}{2} + \frac{1}{2} \right\rfloor = \lfloor i + 1 \rfloor = i + 1,$$

$$\text{RHS} = \left\lceil \frac{k}{2} \right\rceil = \left\lceil i + \frac{1}{2} \right\rceil = i + 1.$$

Thus, LHS $=$ RHS.

In both cases, we have LHS $=$ RHS. Therefore, we conclude that

$$I = \left\lfloor \frac{n}{2} \right\rfloor$$

$\square$

5. *Proof.* Two cases: when n is even and when it is odd.

case 1 $n = 2i$:

$$\begin{aligned}\left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil &= \lfloor i \rfloor + \lceil i \rceil \\ &= i + i \\ &= 2i \\ &= n\end{aligned}$$

case 2 $n = 2i + 1$:

$$\begin{aligned}\left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil &= \left\lfloor i + \frac{1}{2} \right\rfloor + \left\lceil i + \frac{1}{2} \right\rceil \\ &= i + (i+1) \\ &= 2i + 1 \\ &= n\end{aligned}$$

In both cases, we have

$$\left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil = n$$

$\square$

6. The number of internal nodes is $\left\lfloor \frac{n}{2} \right\rfloor$, and the number of leaf nodes is $\left\lceil \frac{n}{2} \right\rceil$.