Binomial Queue

Binomial Queue is a forest of heap-ordered Binomial Tree. A Binomial Tree B_h of height h is recursively defined as:

- B_0 is a single node tree
- B_h is a tree formed by attaching a B_{h-1} to the root of another B_{h-1}

For a given number of elements N, there are a unique sets of binomial trees on the Binomial Queue.

Why the name $Binomial$?

Consider a Binomial Tree B_3 of height 3:

```
1                   level 0, nodes = 1
|   \   \
2    3   4          level 1, nodes = 3
     |   |  \
     5   6   7      level 2, nodes = 3
             |
             8      level 3, nodes = 1
```

The sequence $1, 3, 3, 1$ that represents the number of nodes at each level, are exactly the binomial coefficients. As a recall, binomial coefficients are coefficients of power expansion of the binomial $(x + y)^n$.

The coefficients and thus the number of nodes at each level of a Binomial Tree is given by:

$$\frac{n!}{i! \times (n - i)!} = \frac{height!}{level! \times (height - level)!} = \frac{h!}{i! \times (h - i)!}$$

Hence, the name $Binomial$!!!

Furthermore, the total number of nodes for each Binomial Tree is a function of its height.
Nodes in $B_0 = 2^0 = 1$
Nodes in $B_1 = 2^1 = 2$
Nodes in $B_h = 2^h$

Number of Binomial Trees in a Binomial Queue for a given N is $O(logN)$

Let k be the number of Binomial Trees in a Binomial Queue B containing N elements. When all trees $B_0, B_1, ..., B_h$ are fully occupied we get maximum k. Thus,

$$2^0 + 2^1 + 2^2 + ... + 2^h = N$$

$$2^{h+1} - 1 = N$$

$$h = log(N+1) - 1$$

Height of Binomial Trees in a Binomial Queue is $O(logN)$
Since $k = h + 1$, we get an upper bound on k.

$$k \leq (log(N+1) - 1) + 1$$

As earlier noticed, number of nodes in each binomial tree is a power of 2. It follows, k is also equal to 1's in binary representation of N

```
Examples of Binomial queues:

 B_0  B_1  B_2

  1    2   -1          N = 7 elements  bin(7) = 111 = 2^2 + 2^1 + 2^0
       \    |  \
       3    5  10
            |
            11

 None None -1          N = 4 elements  bin(4) = 100 = 2^2 + 0 + 0
            |  \
            5  10
                |
                11
```