Binomial Heap
(or) Binomial Queue

Definition

- Binomial Queue is a collection of heap-ordered trees, known as forest

- Each of the heap-ordered trees are of a constrained form known as a binomial tree

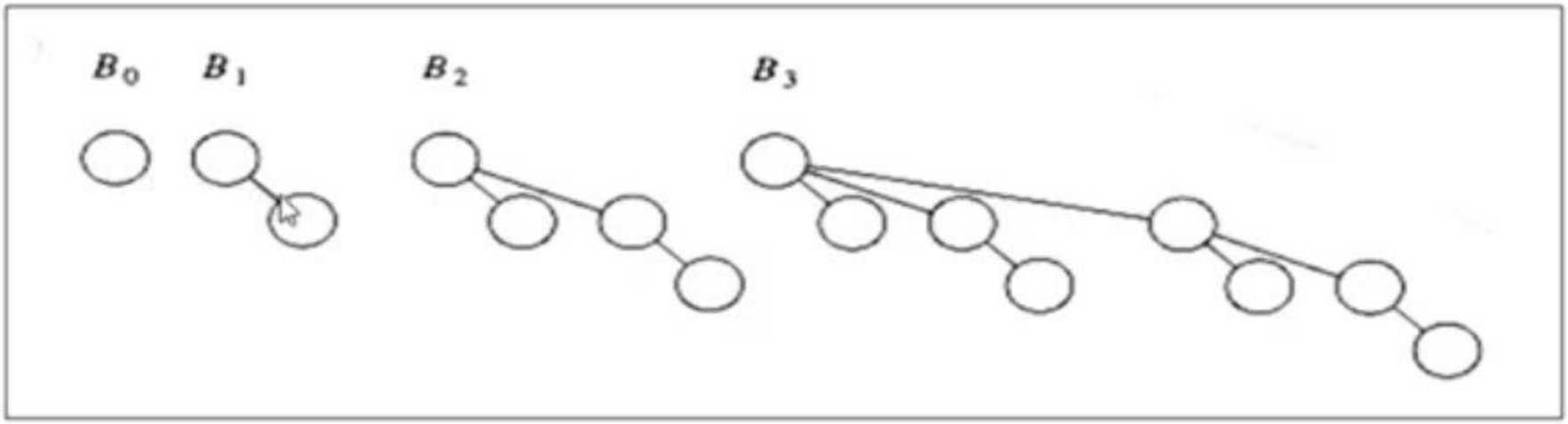
- A binomial tree of height 0 is a one-node tree

- A binomial tree, B_k, of height k is formed by attaching a binomial tree, B_{k-1}, to the root of another binomial tree

Support merge, insert, and deletemin operations in $O(\log n)$ worstcase time.

findMin() in Binomial Queue is performed by searching all the root elements of all binomial trees

There are at most log n different trees, the minimum can be found in **O(log n)** time. Alternatively, we can maintain knowledge of the minimum and perform the operation in O(1) time, if we remember to update the minimum when it changes during other operations.



- Binomial tree, B_0, is one node tree
- Binomial tree, B_1, is formed by merging two B_0 binomial trees
- Binomial tree, B_2, is formed by merging two B_1 binomial trees, selecting one of the nodes as the root node

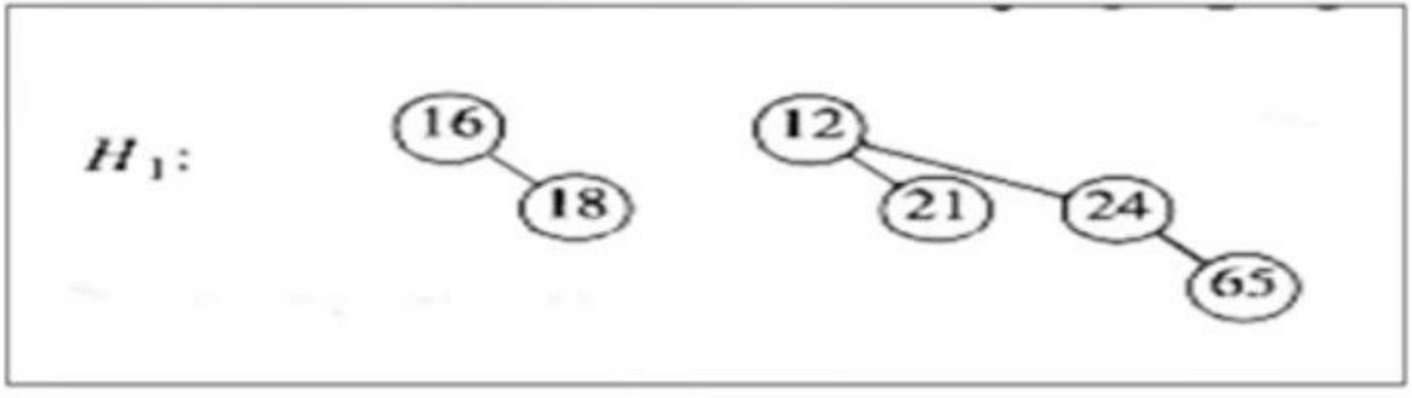
- **There would be 2^k nodes in the binomial tree of height 'k'**
- Binomial tree, B_2 has 4 nodes (2^k); B_3 has 8 nodes (2^3),

Assume priority queue of size 13 (data set size is 13) has to be represented by using Binomial Queue or Binomial Heap.

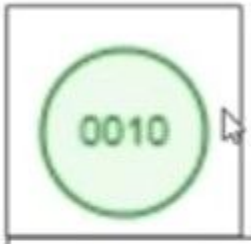
13 is represented as 1101 in the binary format. Hence we require B_3, B_2, B_0 and we don't require B_1.

Example:

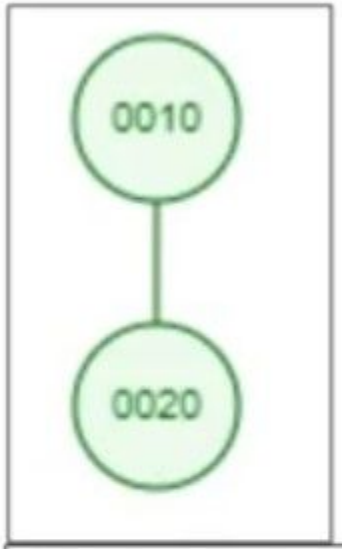
The data 16, 18, 12, 21, 24, 65 are built into binomial heap H1 with Binomial tress B_1 and B_2. Dataset size is 6, (binary is 0110), hence B_1 and B_2



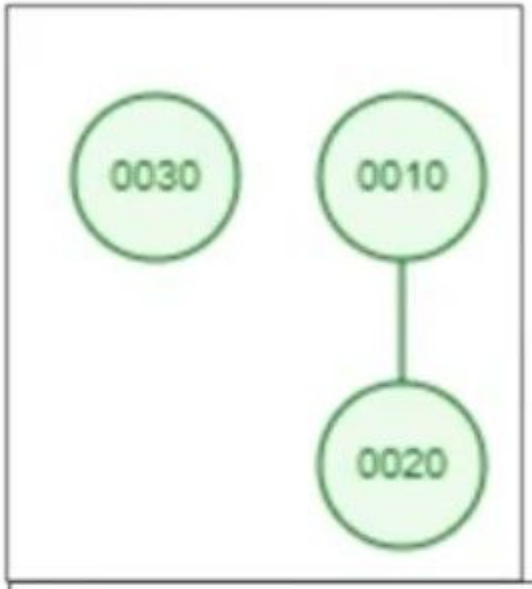
Insert the following items 10. 20, 30, 40, 15, 25 one by one into a Binomial Queue or Binomial Heap



10 is inserted.
There is one item.
So B_0

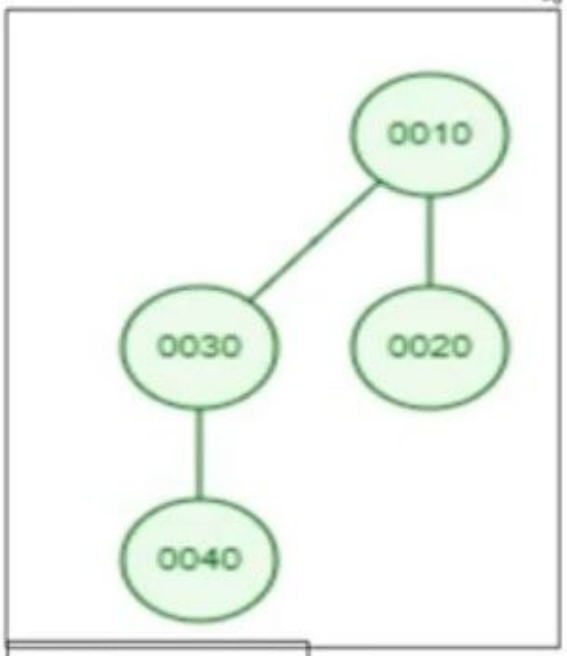


20 is inserted.
There are two items.
So B_1. Two B_0s are merged

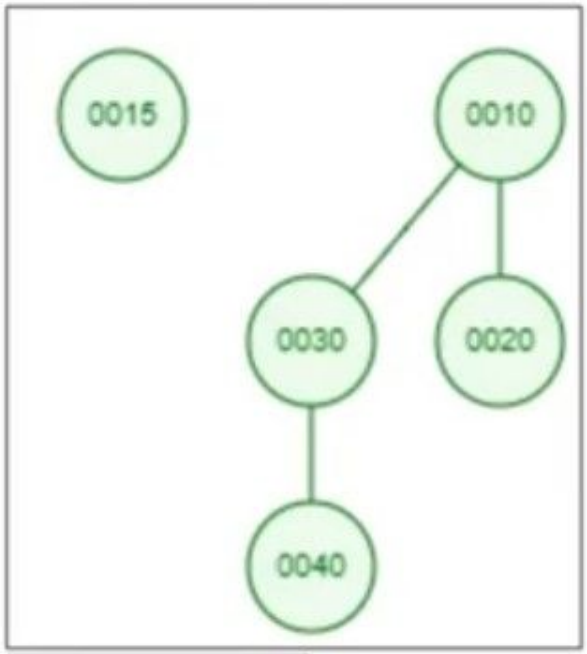


30 is inserted.
There are three items (0011).
So B_0 and B_1.

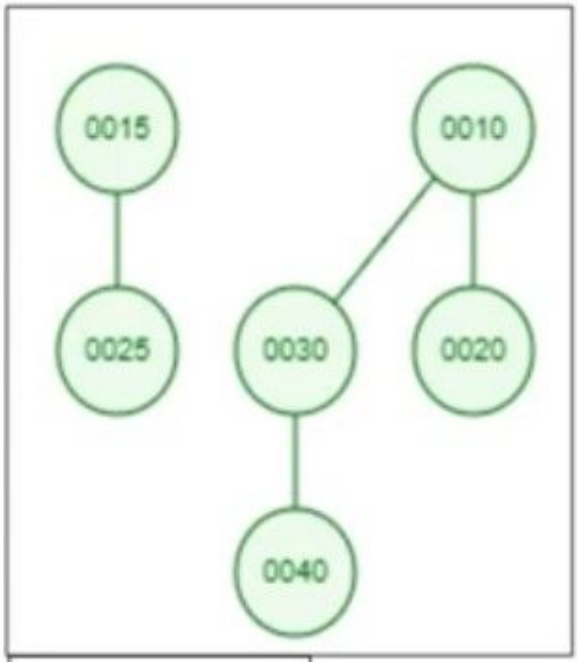
Insert the following items 10. 20, 30, 40, 15, 25 one by one into a Binomial Queue or Binomial Heap



40 is inserted.

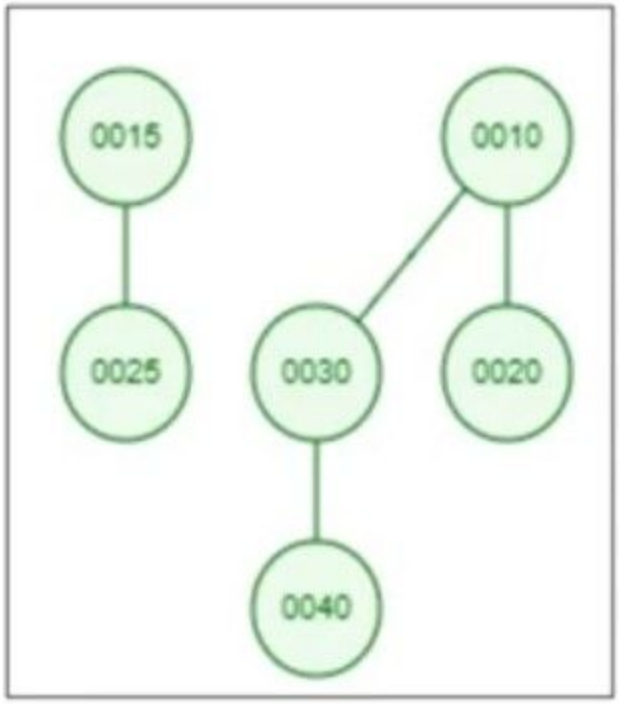


15 is inserted.

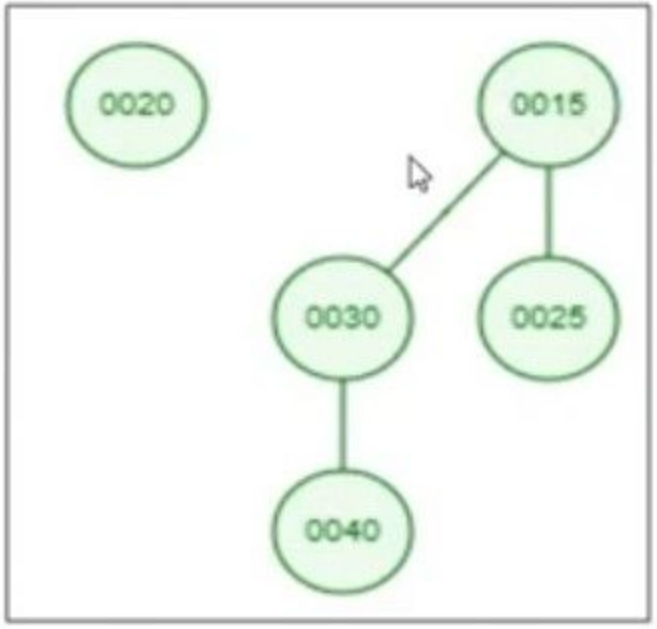


25 is inserted.

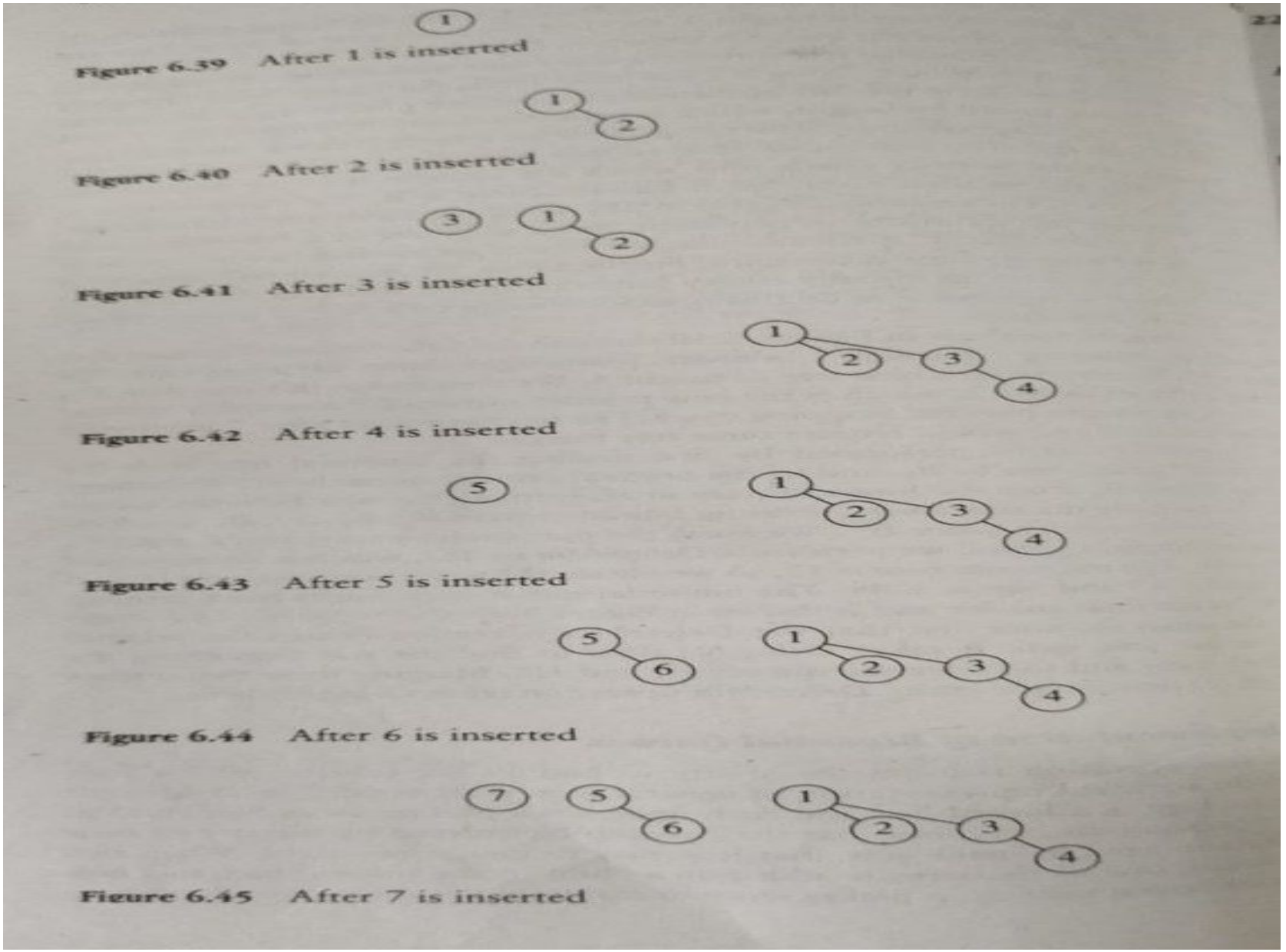
deleteMin() in Binomial Queue is performed by searching all the root elements of all binomial trees, delete the minimum element. Then Rebuild the Binomial Queue



- Root elements of trees are compared (10 and 15)
- 10 gets deleted
- Rebuild the heap ordered trees such that B_0 and B_2 are resulted in the Binomial Heap
- Roots of B_0 and B_2 should have minimum values



Insertion for 1 to 7 numbers



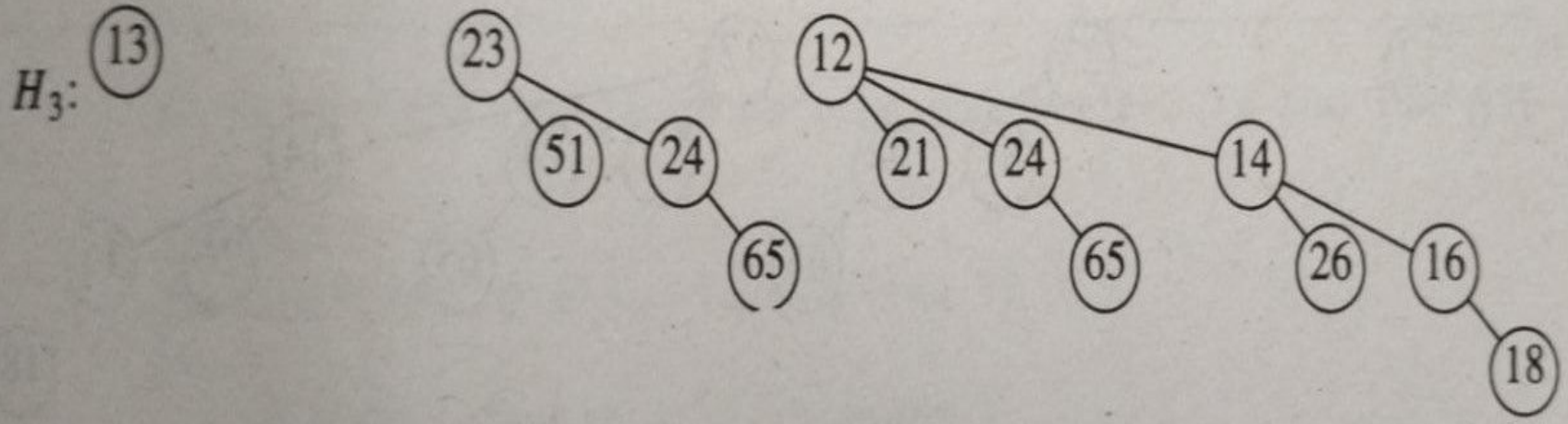


Figure 6.46 Binomial queue H_3

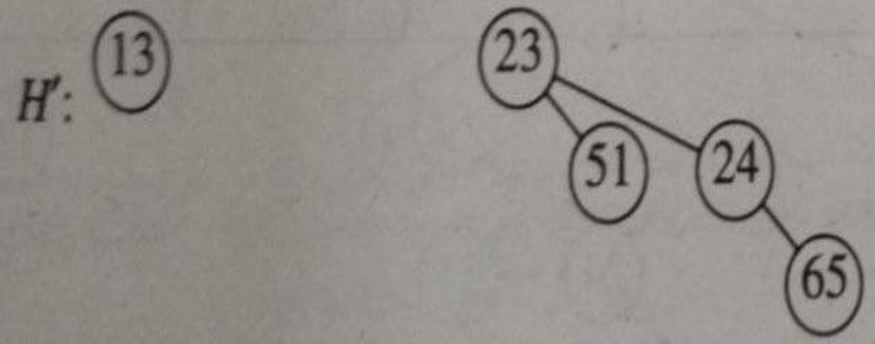


Figure 6.47 Binomial queue H', containing all the binomial trees in H_3 except B_3