Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.4

Aim: Implementation of Binomial Heaps and its various operations

Objective:
- Understanding Binomial Trees , Binomial Heap and its operations.
- Implement Binomial Tree and Binomial Heap class.
- Define the Binomial Heap operations under Binomial Heap class.

Methodology:

I. A Binomial Tree of order 0 has 1 node. A Binomial Tree of order k can be constructed by taking two binomial trees of order k-1 and making one as leftmost child or other.

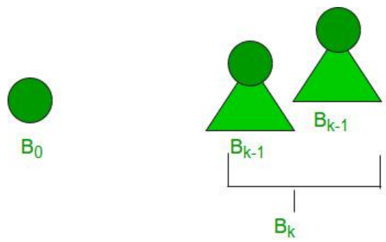
II. A Binomial Tree of order k has following properties:

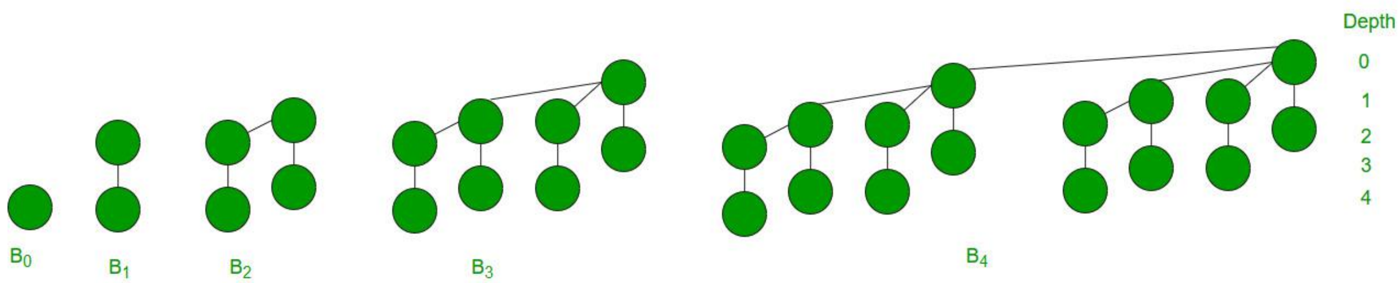
a) It has exactly 2^k nodes.

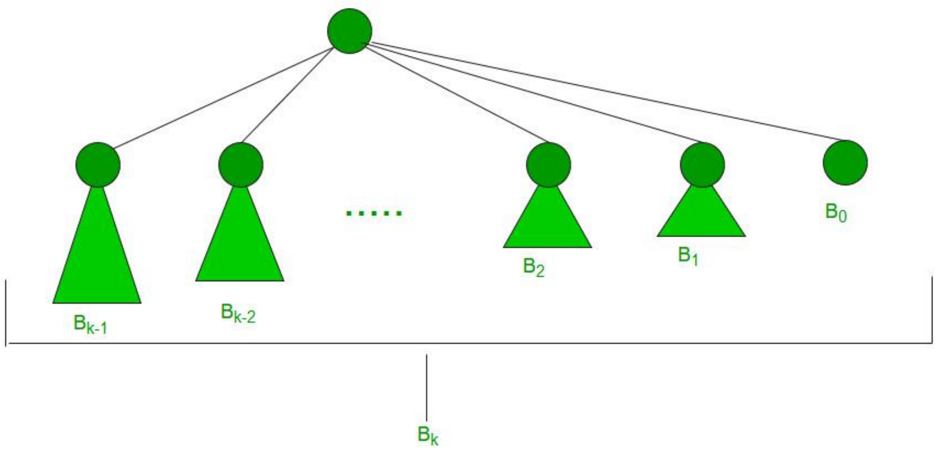
b) It has depth as k.

c) There are exactly ${}^kC_i$ nodes at depth i for i = 0, 1, . . . , k.

d) The root has degree k and children of root are themselves Binomial Trees with order k-1, k-2,.. 0 from left to right.







Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.4

III. A Binomial Heap is a set of Binomial Trees where each Binomial Tree follows Min Heap property. And there can be at most one Binomial Tree of any degree.

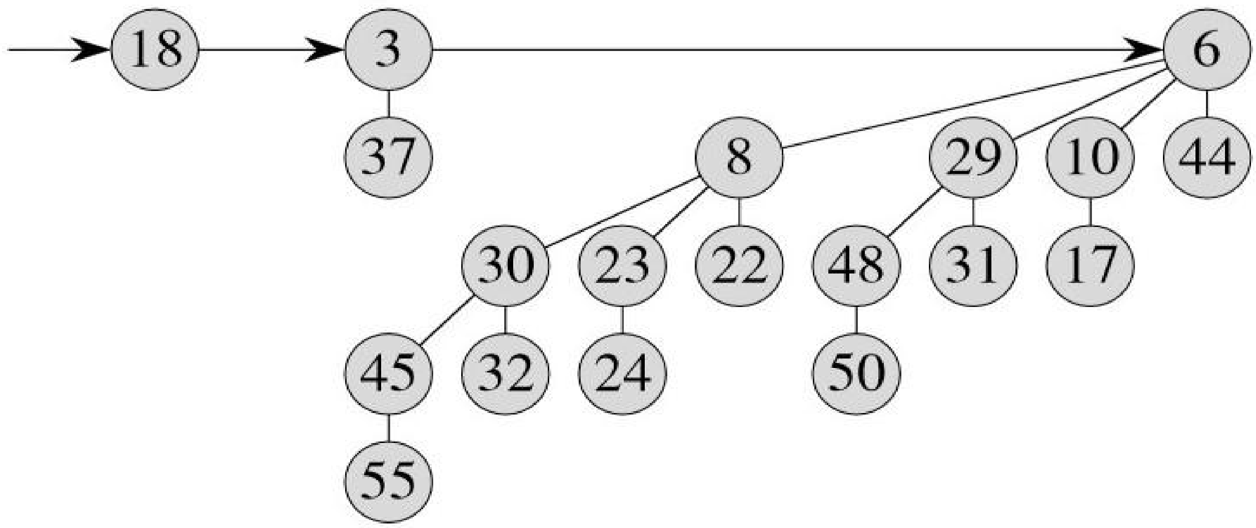
IV. A Binomial Heap with n nodes has the number of Binomial Trees equal to the number of set bits in the Binary representation of n. For example let n be 13, there 3 set bits in the binary representation of n (00001101), hence 3 Binomial Trees.

Implementation:

- Implemented the *class BinomialTree* with following members:
  - *key*
  - *Children (a list)*
  - *order*
- Implemented the *class BinomialHeap* with following member functions:
  - *def extract_min(self)*
  - *def get_min(self)*
  - *def combine_roots(self, h)*
  - *def merge(self, h)*
  - *def insert(self, key)*
- Mainly 3 operations are implemented by User view named:
  - Insert.
  - min get.
  - min extract.

Results:

Input:



Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.4

Output:

PROBLEMS OUTPUT TERMINAL 1: Code

```
C:\Users\Vishal Ramane\OneDrive\College\AAC Lab\Code\Expt4>python -u "c:\Users\Vishal Ramane\OneDrive\College\AAC Lab\Code\Expt4\BinomialHeap.py"
 ==========: Menu ==========:
insert <data>
min get
min extract
quit
What would you like to do? insert 18
What would you like to do? insert 3
What would you like to do? insert 37
What would you like to do? insert 6
What would you like to do? insert 8
What would you like to do? insert 29
What would you like to do? insert 10
What would you like to do? insert 44
What would you like to do? insert 30
What would you like to do? insert 23
What would you like to do? insert 22
What would you like to do? insert 48
What would you like to do? insert 31
What would you like to do? insert 17
What would you like to do? insert 45
What would you like to do? insert 32
What would you like to do? insert 24
What would you like to do? insert 50
What would you like to do? insert 55
What would you like to do? min get
Minimum value: 3
What would you like to do? min extract
Minimum value removed: 3
What would you like to do? min extract
Minimum value removed: 6
What would you like to do? min extract
Minimum value removed: 8
What would you like to do? min extract
Minimum value removed: 10
What would you like to do? min get
Minimum value: 17
What would you like to do? min extract
Minimum value removed: 17
What would you like to do? min get
Minimum value: 18
What would you like to do? quit

C:\Users\Vishal Ramane\OneDrive\College\AAC Lab\Code\Expt4>
```

Ln 78, Col 39 Spaces: 4 UTF-8 CRLF Python

Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.4

Conclusions:
Thus we have successfully implemented Binomial Heap and its various applications.