Binomial Heap (Write-up)

Amber Mishr
1BM18CS0

```
class Node
{  int data, deg;
     Node child, sibling, parent;
 }.
Nodode newNode (int k)
 {
    Node temp = new Node();
      t.data = k;
      t.deg = 0;
     t.child = t.sibling = t.parent = NULL;
     return t;
   }.
  List <Node> insert ( head, key)
    {     Node t = new Node (key)
            List temp;
             temp.add (t)
             temp = union Binomial (head, temp);
             return adjust (temp);
```

Ambuj Mishra
1BM18CS023

```
List <Node>    unionBinomial (List l1, List l2)
{.
    List <Node> new;

    while (l1!=null && l2!=null)
    {.
        if ( l1.deg <= l2.deg)
        {
            new.add(l1)
            l1 = l1.next.
        }
        else
        {   new.add (l2)
            l2 = l2.next;
        }
    }
    while (l1!=null)
    {   new.add (l1)
        l1 = l1.next;
    }
    while (l2!=null)
    {   new.add(l2)
        l2 = l2.next;
    }.
    return new
}.
```

Ambar Mishra
1BM18CS013

```
List <Node> adjust ( List <Node > heap)
{
        if (heap.size() <= 1)
              return heap.
        List new_heap;
        List <Node> iterator  it1, it2, it3;
         if (heap.size() == 2)
          {   it2 = it1
              it2++;
              it3 = heap.end();
          }.
         ~~else~~ else
          {     it2++;
                it3 = it2
                it3++;
             }
        while (it1! = null)
         {
             if (it2 == null)
                 it1++;
             else if (it1.deg < it2.deg )
             { it1++
               it2++
```

Scanned with CamScanner

Amlan Mishra
1BM18CS013

```
    if (it3! = null)
        it3 ++;
    }.
else if (it3! = null && it1.deg == it2.deg && it1.deg
            == it3.deg)
    {
        it1++;
        it2++
        it3++
    }.
    return head;
}.
Node getMin (List head)
 {
    List <Node> it;
    Node temp = it;
    while (it! = null)
     {
        if ( it.data < temp. data)
            temp = it
        ~~ra~~ it = it.next
     }
     return temp;
  }.
```

Scanned with CamScanner

Amber Mishra
1BM18CS013

```
List <Node> extractMin (List <Node> heap)
{   List <Node> new_heap, lo;
      Node temp;
    temp = getMin(_heap)
    List<Node> it;
     while (it != null)
      {     if (it != temp )
                new_heap.add(it)
            it = it.next();
      }
         new_heap = unionBinomial (new_heap, lo)
            new_heap = adjust (new_heap)
              return new_heap;
      }
}
```