Rutazeet Ritik Rout
1BM18CS151
9|12|20

Advanced Data Structure
Batch - 5

Program ~ 9

```
Insert (heap, value){
    create new node with value or key as value
    create temporary heap
    looping over heaps until it becomes NULL:
            if degree of original tree in heap
    is less than degree of temporary tree in heap
                Create new heap and add original tree
        else
                add temporary tree to heap
      if original heap has left over tree.
                add them to new heap

      if temporary heap has left over tree
                add all of them to new heap.

      if heap size <1 return heap

      loop over new heap
            if it ends of heap
                    One element remains.
            else if degree first tree less than degree
              of second tree,
                      merge.
          else if degree are same then
                  binomial tree are same in heap.

      return heap.
}
```

Rutazeet

2

1BM18CS151

```
get min (heap) {
    Start from 1st tree in heap & check root of tree.
    find min of all roots and return.
}

Extract min ( heap) {
  get min value by function getmin ()
  Start from first tree in heap.
  if tree root is not minimum then create new heap.
     & add tree to heap

  remove minimum ele from heap & convert tree to heap.

  Merge newly created heap without mini. element
  & heap that was created earlier.

  return merged heap
}.
```