9-12-20

ADS lab

B Pranecth
IBM18CS023

Program 9

Binomial Heap

(1) insert(H,k)

```
H' <- MAKE-BINOMIAL-HEAP() //new node or value to be inseted
P[x] <- NIL
child[x] <- NIL
sibling[x] <- NIL
degree[x] <- 0
head[H'] <- x
H <- BINOMIAL-HEAP-UNION(H,H') //merge the newly
                               //created BO tree with
                               //already existing tree
```

(2) getMin(H)

```
*iterator = H.begin()
*temp = *iterator
while (iterator != heap.end()){
        if ((iterator->data < temp->data)
                temp = *iterator
        iterator++;
}
return temp;
```

// Traverse the list of root of
// Binomial Trees and return the
// minimum key

BPranecth

Program 9 - Binomial Heap

(3) extractMin(H)

```
// First call getMin() to find minimum key, then remove
// the Node and create a new Binomial Heap by connecting
// all subtrees of the removed minimum node.
H' ← MAKE-BINOMIAL-HEAP
it = head.begin()
while (it != head.end())
        if (*it != temp)
          {
            H'.push_back(*it)
          }
        it++;
  }
lo = removeMinandReturnBHeap(temp);
new_heap = union(H', lo)
new_heap = adjust(H')
 return H';
}
```


B Praneeth