Neelesh Gangopadhyay
1BM18CS058
5th Sem B

SURYA Gold Date Page

Program 16

Binomial Heap Decrease key (u)
Delete (M)

```
void decreaseKeyBin (Node * H, int old val
                     int new val)
{

~~Node * node~~
  // 1. Check element is present or not
  // 2. Return of the node is not present
  // 3. Reduce value to minimum
  // 4. Update the heap accordingly
  //    to the reduced value
  //

Node * node = findNode (H, old val)
    if (node == NULL)
        return;
node -> val = new_val;
Node * parent = Node -> parent;

while (parent != NULL && node -> val)
{
  Swap (node -> val, parent -> val).
       node = parent
  parent = parent -> parent;
    }
// Function Deletion Element from Heap
```

```
Node * binDelete (Node* h, int val)

{  // Check if heap is empty or not
// 2 Reduce value to minimum
  // 3. Delete min Element from BHeap

   if (h == NULL)
           return NULL;
decrease KeyBHeap (h, val, INT_MIN);
return extractMin (h);
}
Node * FindNode (Node* h, int val)
  {    if (h == NULL)
           return NULL
       if (h -> val == val)
           return h;

   Node * res = findNode (Node* h, int val)
    {  if (h == NULL)
           return NULL;
       if (h -> val == val)
           return h;
    Node* res = findNode(h -> child, val);
           if (res != NULL) return res;

    return findNode(h -> sibling, val)
      }
```