Akanksha Laddha
1BM18CS007

ADS - Lab 9.

Batch B1.

Date: December 9, 2020
Section: 5A.

Program to implement binomial heap.

```
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 lesser than degree of temporary tree in heap
            create new heap & add
            original tree
        else
            add temporary tree to heap.
    if original tree/heap had left over heap 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 its end of heap
            one elem. remains
        else if degree first tree less than
        degree of second tree,
            merge.
```

Akanksha

Date:
P. No:

else if degree are same then
binomial tree are same in heap.

return heap.

}

getmin (heap)
{

start from 1st tree in heap & check
root of tree. find min. of all roots
& return.

}.

Extract min (heap)
{

get min value by func" 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.

}.

Kavita

Scanned by TapScanner