

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					
M	T	W	T	F	S

Aubay Ahmed  
IBM IQCS401

# Binomial heap

(i) insert (H, k)

(ii) getMin(H)

(iii) extractMin(H)

## struct node

```

int data;
Node *child, *sibling, *parent;
int degree;

```

```

Node *newNode (int key)

```

```

Node *temp = new Node;
temp->data = key;
temp->degree = 0;
temp->child = temp->parent = temp->sibling = null;
return temp;

```

```

int insert (Node *head, int key)
{
    Node *temp = newNode(key);
    return insertBintree(head, temp);
}

```



OCTOBER 2016  
TUESDAY

30	31					
2	3	4	5	6	7	1
9	10	11	12	13	14	8
16	17	18	19	20	21	15
23	24	25	26	27	28	22
Su	M	T	W	T	F	5

WK 44 - 299-067

Node \* getMin (int &Node \* & heap)

8. In a node  $x$ , level  $x$  is less than  $x$  is less than  $x$ .

~~Node temp = p~~

10 While (P != head.end()) {

17 (nit)  $\rightarrow$  date  $\angle$  temp  $\rightarrow$  date

temp = 40°C

12

2 return temp

list (Node\*) &insertMen (list (Node\*) head)

but  $\text{node} \rightarrow \text{next} \rightarrow \text{node}$ ;

Node \* temp;

temp = getLen (hush);

let  $\{nodes\}$  ::  $list$  of  
 $uf = head(nodes)$

if = head (begin)

While (it != head, end()) {

if  $\text{tail} \geq \text{temp}$

new high fresh back (set)

y (4700-1700)



1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					
M	T	W	T	F	S

2016 OCTOBER  
WEDNESDAY

26  
300-066 WK 44

lo = remove Node from Tree Return Bst heap (temp);  
new heap = union of new heap (new heap, lo);  
new heap = adjust (new heap);  
return new heap;

void printTree (Node \*h)

{ while (h) {  
    cout << h->data << " ";  
    printTree (h->lchild);  
    h = h->rchild;  
}

void printHeap (int \*Node \*h) {

    int \*Node \*p; dealer p;  
    p = -heap begin (1);  
    while (p != heap end())  
    { printTree (\*p);  
      p++;  
    }

}