

## LAB 9. Binomial Heap

Nikhil A.S  
IBM18(CS06)

9/10/2020

Node \* mergeBinomialTrees(Node \*b1, Node \*b2)

{ if ( $b1 \rightarrow \text{data} > b2 \rightarrow \text{data}$ )  
swap( $b1, b2$ );

$b2 \rightarrow \text{parent} = b1$

$b2 \rightarrow \text{ sibling} = b1 \rightarrow \text{child}$ .

$b1 \rightarrow \text{child} = b2$

$b1 \rightarrow \text{degree}++$

return  $b1$

List<Node\*> UnionBinomialHeap(List<Node\*> l1, List<Node\*> l2)

{ List<Node\*> \_new  
List<node\*> :: iterator it = l1.begin();  
List<node\*> :: iterator ot = l2.begin();  
List<node\*> :: iterator o  
while (it != l1.end() && ot != l2.end())  
{ if ((\*it)  $\rightarrow$  degree  $\leq$  (\*ot)  $\rightarrow$  degree)  
{ \_new.push\_back(\*it);  
it++;  
}  
else  
{ \_new.push\_back(\*ot);  
ot++;  
}  
}

```
    while(it != ll.end())
    {
        new.push_back(*it);
        it;
    }
```

```
    while(ot != ls.end())
    {
        new.push_back(*ot);
        ot++;
    }
```

```
list<Node*> adjust(list<node*> -heap)
{
    if(-heap.size() <= 1)
        return -heap;
```

```
    list<Node*> new-heap;
    list<Node*> :: iterator it1, it2, it3;
    it1 = it2 = it3 = -heap.begin();
```

```
    if(heap.size() == 2)
```

```
    { it2 = it1;
```

```
        it2++
```

```
        it3 = -heap.end();
```

```
}
```

```
else
```

```
{ it2++;
```

```
    it3 = it2
```

```
    it3++;
```

```
}
```

```

while (it1 != -heap.end())
{
    if (it2 == -heap.end())
        it1++;

    else if ((*it1) > degree < (*it2) > degree)
    {
        it1++;
        it2++;
        if (it3 != -heap.end())
            it3++;

        else if (it3 != -heap.end() && (*it1) > degree && (*it2) > degree && (*it3) > degree)
        {
            it1++;
            it2++;
            it3++;
        }
        else if ((*it1) > degree == (*it2) > degree)
        {
            Node *temp;
            *it1 = mergeBinomialTrees(*it1, *it2);
            it2 = heap.erase(it2);
            if (it3 != -heap.end())
                it3++;

            else
                }
        }
    }
}

```