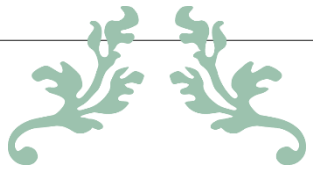

FIBONACCI HEAP

COP5536-Advanced Data Structures

8971-8937
DAYE KIM
dayekim@ufl.edu

FIBONACCI HEAP

8971-8937 · Da Ye Kim
dayekim@ufl.edu

ESSENTIAL FLOW OF FIBONACCI HEAP

1. Detect '#' and read the string[name] and the int[key] from the input-file
2. Create new node and store this in hashmap with the name
   ,if there is not the same node with the name.
   Or, Add up the key from input and the key of the node whose name is the same with the name of input string[name]
3. Performs increaseKey operations in Fibonacci Heap

1. Detect int[digit]
2. Extract the digit number of maximum values from Fibonacci Heap and hashmap
3. Meld the children nodes with the nodes whose degree are the same after extracting parent nodes[maximum nodes]
4. Print the name of the extracted nodes on command window or write on output file
5. Insert the extracted nodes in Fibonacci Heap and hashmap again.

NODES

```
int key;
// node's key value
//stored in hashMap(int)

private String name;
//node's name = input string[name]
//stored in hashMap(string)

boolean mark = false;
//to check if this node lose the child node.
```

//if it lost child node, it turns to True

int degree = 0;
//the height of nodes

Node child, parent, left, right;
//each node has child, parent, left, right nodes.

FIBONACCI HEAP

private Node maximumNode;

private int totalNumNodes;

- Remove Maximum Node
- Insert
- Increase Key
- Cut / cascading
- Meld (samedegreeMeld)
- Generate Child Node

DESCRIPTIONS

- Remove Maximum Node – `removeMax()`

  1. Remove children Nodes from Maximum Node.
  2. Maximum Node is removed from root list.
  3. Meld the children trees by the same degree of the top nodes of children trees (using 'samedegreeMeld')

- Insert Node – `insert(Node node)`

  1. Check if the node is maximum value, and insert the node in roots in Fibonacci Heap After removing Maximum Node or after creating New node
  2. If there is not maximum value, the input node is the maximum node.

- Incerese Key- `increaseKey(Node a, int vkey)`
  1. Node a's key is changed to the value of 'vkey' which is increased by adding up in hashtagcounter.
  2. The node whose key is increased is `cut()` and `casCading ()`
  3. If the node's key is maximum, change the maximum node to this node
- Cut/ cascading- `cut(Node a, Node b) / casCading(Node b)`
  1. Cut node a (child) from node b(parent)
  2. Decrease the degree of b
  3. Move a into roots in Fibonacci Heap
  4. Perform -`casCasding(node b)`
  5. In cascading operation, the node b's mark is changed true since it lost child, and perform cascading with b's parent node.
- Meld- `samedegreeMeld()`
  1. Check all degree of the nodes in roots and put into the degreeRangeTable
  2. If there is the same degree in the table, compare values of each key and meld, using `generateChild(Node ch, node a)`
- Generate Child node - `generateChild(Node ch, Node a)`
  1. Node ch will be child node of node a
  2. Node ch is removed from other nodes and attached to node a
  3. Increase node a's degree

HASHTAGCOUNTER

In this class, all the operations perform.

```
FibonacciHeap fnc = new FibonacciHeap();
```

```
HashMap<String,Node> hsm = new HashMap();
```

```
Node node = new Node(name,key);
```

With two objects, Fibonacci Heap is embodied.

```
Pattern count = Pattern.compile("(\\d+)");
Matcher ct = count.matcher(ph);
```

Detect pattern int[digit] – the number of maximum nodes to be extracted.

```
Pattern htag = Pattern.compile("([#])([a-z_]+)(\\s)(\\d+)");
Matcher hg = htag.matcher(ph);
```

Detect pattern: '#' and read the string[name] and the int[key] from the input-file

1. If the name[classified in group(2) in hg] is matched with the name in hashMap,
   Adds up the values of the keys and performs IncreaseKey()
   If not, Create new node and insert() the node in Fibonacci Heap and hashMap
2. If the digit[classified in group(1) in ct] is read,
   Performs removeMax() and extracted nodes are stored in maxNumNodes arraylist.
   Create new nodes of extracted nodes and insert() in Fibonacci Heap and hashMap again after printing the nodes on command window.