SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Department of Information Technology

COURSE CODE: DJS22ITL502 **DATE: 7-10-2024**

COURSE NAME: Advanced Data Structures Laboratory **CLASS:** TY B. TECH

NAME: Anish Sharma **DIV:** IT 1 **ROLL:** I011

EXPERIMENT NO. 4

CO/LO: Choose appropriate data structure and use it to design algorithm for solving a specific problem

AIM / OBJECTIVE: To implement various operations on Fibonacci Heap.

Properties of Fibonacci Heap:
Amortized Efficiency: Insertions and decrease-key operations take O(1) amortized time.
Lazy Merging: Trees are merged lazily, with restructuring delayed until necessary.
Find Min: Finding the minimum element takes O(1) time by accessing the min pointer.
Consolidation on Extract Min: Extracting the minimum takes O(logn) by consolidating trees.
Decrease Key: Decreasing a key cuts the node and its subtree, and inserts it into the root list in O(1) amortized time.

TECHNOLOGY STACK USED: C, C++, JAVASOURCE

CODE:

```
import java.io.*;

class Node {
    Node parent;
    Node child;
    Node left;
    Node right;
    int key;
}
class Main {
    static Node mini = null;
    static int no_of_nodes = 0;
    static void Insertion(int val)
    {
        Node new_node = new Node();
        new_node.key = val;
        new_node.parent = null;
        new_node.child = null;
        new_node.left = new_node;
        new_node.right = new_node;

        if (mini != null) {
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Department of Information Technology

```
            (mini.left).right = new_node;
            new_node.right = mini;
            new_node.left = mini.left;
            mini.left = new_node;

            if (new_node.key < mini.key)
                mini = new_node;
        }
        else {
            mini = new_node;
        }
    }
    static void Display(Node mini)
    {
        Node ptr = mini;
        if (ptr == null)
            System.out.println("The Heap is Empty");

        else {
            System.out.println(
                "The root nodes of Heap are: ");
            do {
                System.out.print(ptr.key);
                ptr = ptr.right;
                if (ptr != mini) {
                    System.out.print("-->");
                }
            } while (ptr != mini && ptr.right != null);
            System.out.println();
            System.out.println("The heap has " + no_of_nodes
                               + " nodes");
        }
    }
    static void FindMin(Node mini)
    {
        System.out.println("min of heap is: " + mini.key);
    }

    public static void main(String[] args)
    {
        no_of_nodes = 7;
        Insertion(4);
        Insertion(3);
        Insertion(7);
        Insertion(5);
        Insertion(2);
        Insertion(1);
        Insertion(10);
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Department of Information Technology

```
        Display(mini);

        FindMin(mini);
    }}
```

OUTPUT:

```
The root nodes of Heap are:
1-->2-->3-->4-->7-->5-->10
The heap has 7 nodes
min of heap is: 1
```

CONCLUSION: In this experiment we understood and implemented Fibonacci heaps

REFERENCES:

1. Peter Brass, "Advanced Data Structures", Cambridge University Press, 2008
2. Robert Sedgewick & Kevin Wayne, "Algorithms", 4th Edition, Addison-Wesley Professional, 2011.