

IIT Roorkee

CSN 261:Data Structures

Laboratory



Assingment 6

Name:Ritesh Singh

Enrollment Number:18114067

Batch-O3

Email: rsingh1@cs.iitr.ac.in

Problem 1 :

Write a menu driven C++ program to implement a graph using adjacency list (linked list) without using STL. Perform following operations on the graph.

1. Inset edge
2. BFS traversal
3. DFS traversal
4. Cycle finding in the graph
5. Calculate diameter of the graph

Algorithms Used :

1. Breadth First Search
2. Depth First Search
3. Depth First Traversal can be used to detect a cycle in a Graph. DFS for a connected graph produces a tree. There is a cycle in a graph only if there is a back edge present in the graph. A back edge is an edge that is from a node to itself (self-loop) or one of its ancestor in the tree produced by DFS.

Data Structure Used :

1. Arrays
2. Linked Lists
3. Vectors
4. Maps

Snapshots :

```
1. Inset edge
2. DFS traversal
3. BFS traversal
4. Cycle finding in the graph
5. Calculate diameter of the graph
6. Exit
2
A      B
A      C
F      D
D      C
U      I
U      D
B      I
F      B
7
A B I U D F C
```

```
1. Inset edge
2. DFS traversal
3. BFS traversal
4. Cycle finding in the graph
5. Calculate diameter of the graph
6. Exit
3
A      B
A      C
F      D
D      C
U      I
U      D
B      I
F      B
7
A B C I F D U
```

```
1. Inset edge
2. DFS traversal
3. BFS traversal
4. Cycle finding in the graph
5. Calculate diameter of the graph
6. Exit
5
A      B
A      C
F      D
D      C
U      I
U      D
B      I
F      B
7
Diameter of the given tree is 7
```

```
1. Inset edge
2. DFS traversal
3. BFS traversal
4. Cycle finding in the graph
5. Calculate diameter of the graph
6. Exit
4
A      B
A      C
F      D
D      C
U      I
U      D
B      I
F      B
7
Graph contains cycle
```

```
real    0m2.386s
user    0m0.006s
sys     0m0.001s
```

Problem 2 :

A binomial heap is implemented as a set of binomial trees, which are defined recursively as follows:

- x A binomial tree of order 0 is a single node
- x A binomial tree of order k has a root node whose children are roots of binomial trees of orders $k-1, k-2, \dots, 2, 1, 0$ (in this order).
- x A binomial tree of order k has 2^k nodes, height k .

Write a C++ program to implement a binomial heap using heap data structures (without using STL).

Print the order of each binomial heap and use Graphviz to show the forest of binomial heap.

Algorithms Used :

A Binary Heap is a Binary Tree with following properties.

- 1) It's a complete tree (All levels are completely filled except possibly the last level and the last level has all keys as left as possible). This property of Binary Heap makes them suitable to be stored in an array.
- 2) A Binary Heap is either Min Heap or Max Heap. In a Min Binary Heap, the key at root must be minimum among all keys present in Binary Heap. The same property must be recursively true for all nodes in Binary Tree. Max Binary Heap is similar to MinHeap.

Data Structure Used :

1. Linked List

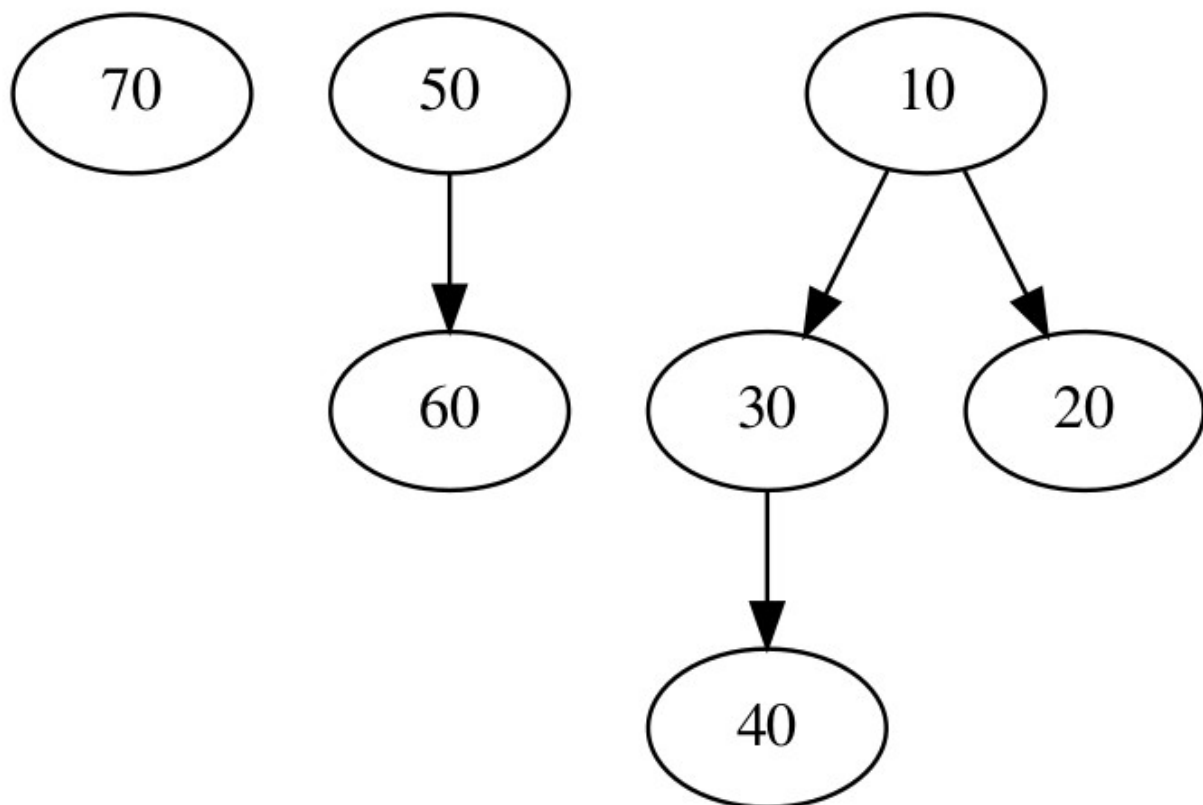
2. Structures

Snapshots :

```
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 10
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 20
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 30
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 40
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 50
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 60
1)Insert Element in the heap
2)Display Heap
3)Exit
Enter Your Choice: 1
Enter the element to be inserted: 70
```

```
1)Insert Element in the heap  
2)Display Heap  
3)Exit  
Enter Your Choice: 2  
The Heap is:  
0:70  
1:50 60  
2:10 30 40 20
```

```
real    0m14.671s  
user    0m0.007s  
sys     0m0.000s
```



Problem 3 :

Write a C++ program to implement Bentley-Ottmann Algorithm to find and print all the intersection points of n given lines. Use of STL is allowed. The specific type of data structure that must be used include Priority Queue and BST. Using least square method find the linear fit of the M found intersection points and print the line in the form $ax+b$. The student should demonstrate this on a GUI using QT library. The input should be given in following format:

1. Input number of line segments, N
2. N lines where $2N$ points are provided, i.e., 2 points in each line

Algorithms Used :

1. Bentley-Ottmann Algorithm

Data Structure Used :

1. Priority Queue

2. BST

3. List

Snapshots :

```
N=6
104 212 513 727
229 424 538 278
249 324 654 657
508 440 531 623
453 295 517 398
639 290 601 116
No. of intersection points: 4
(260.53, 409.10)
(318.94, 381.50)
(464.13, 312.91)
(521.59, 548.13)
Linear fit: 0.2937x + 297.9693

real    0m6.909s
user    0m0.005s
sys     0m0.000s
```