**The laboratory work 6**

**Breadth-first search** is a simple strategy in which the root node is expanded first, then all the successors of the root node are expanded next, then *their* successors, and so on. In general, all the nodes are expanded at a given depth in the search tree before any nodes at the next level are expanded. Breadth-first search is an instance of the general graph-search algorithm in which the *shallowest* unexpanded node is chosen for expansion. This is achieved very simply by using a FIFO queue for the frontier.

The binary tree traversal will look like this.



Assignment – write a program code that will realize the traversal of the binary search tree. In the first part, we need to create a class of the binary tree that will keep data about the structure of the binary tree.

using namespace std;

class Node {

int data;

Node \*left;

Node \*right;

public:

Node() { data = -1; left = NULL; right = NULL; };

void setKey(int key) { data = key; };

void setLeft(Node\* Left) { left = Left; };

void setRight(Node\* Right) { right = Right; };

int getKey() { return data; }

Node \*getLeft() { return left; }

Node \*getRight() { return right; }

}

Then you will need to create addNode function to insert a new node in the binary tree and levelOrder to traverse the binary tree as the breadth first search.

int main()

{

Node \*tree = new Node();

tree->addNode(5, tree);

tree->addNode(4, tree);

tree->addNode(36, tree);

tree->addNode(3, tree);

tree->addNode(34, tree);

tree->addNode(8, tree);

tree->addNode(1, tree);

cout << "The binary tree we have is " << endl;

tree->levelOrder(tree);

return 0;}