American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-10, Issue-6, pp-50-53
www.ajer.org
Research Paper Open Access

Equation tree or ABL tree to represent data with high security

Binu C T
Software Test Engineer
Syntrio technologies, technopark, Thiruvananthapuram

ABSTRACT: *Data structure is the way to represent data in our system. Data structure are two different types Linear and nonlinear. If we represent both linear and nonlinear into a tree, linear have one successor but nonlinear have more than one successor. Tree in data structure is a most important type of nonlinear data structure in computer science. There are a lot of trees like binary search tree, heap tree, Red black tree etc. The proposed tree uses linear equation or binomial equation to map the data into a tree. The proposed tree is called hash tree or equation tree. We can use different equations to represent data in different systems which increase the security of the system. The time complexity of the proposed algorithm is O (n)*
KETWORDS: *Tree, Binary search tree, Integrity, Confidentiality, Security*

Date of Submission: 25-05-2021 Date of acceptance: 07-06-2021

I. INTRODUCTION

The tree consists of nodes or *vertices* and *edges* or lines. Nodes are usually labelled with data or search key. The top level of the tree is called *roots.* The predecessor of the node is called *parent* and next level node is called *siblings.* Last nodes are called leaf nodes. *Depth* of the tree means the length at which we reach the leaf node.

II. EQUATION TREE OR ABL TREE ALGORITHM

Read the length of the tree n
Read number of equation l
Select linear or binomial equation
Read *data*
Read the linear or binomial equations
Put 0 to variable in the root equation x
Leaf(v) = MakeTree(v; EmptyTree; EmptyTree)
MakeTree(2, MakeTree(6,MakeTree(15,EmptyTree,EmptyTree),
MakeTree(23,EmptyTree,MakeTree(46,EmptyTree,EmptyTree)))
,
MakeTree(75,MakeTree(9,EmptyTree,MakeTree(98,EmptyTree,EmptyTree)),

MakeTree (data,left,right)
If(data<x)
Root=*data*
y=put depth to the variable of left equation
if(data<y)
compare *data* with existing nodes
left(node)=*data*
z= put depth to the variable of right equation
if(data<y)
compare *data* with existing nodes
right(node)=*data*

III. TIME COMPLEXITY

Read the length of the tree n	1
Read number of equation l	1
Select linear or binomial equation	1
Read *data*	1
Read the linear or binomial equations	1
Put 0 to variable in the root equation x	1
Leaf(v) = MakeTree(v; EmptyTree; EmptyTree)	n
MakeTree(2, MakeTree(6,MakeTree(15,EmptyTree,EmptyTree),	n
MakeTree(23,EmptyTree,MakeTree(46,EmptyTree,EmptyTree))),	n
MakeTree(75,MakeTree(9,EmptyTree,MakeTree(98,EmptyTree,EmptyTree)),	n
MakeTree(data,left,right)	
If(data<x)	1
Root=*data*	1
y=put depth to the variable of left equation	1
if(data<y)	
compare *data* with existing nodes	1
left(node)=*data*	
z= put depth to the variable of right equation	1
if(data<y)	
compare *data* with existing nodes	1
right(node)=*data*	1

The time complexity of the proposed algorithm is O (n)

Example:

Root equation =5x+3

Left equation=3x+1

Right equation=4x+1

2,6,15,23,46,75,98

Fig.1

The first data is consider as root node and compare with value of equation of root. Here I put the value of the variable in the equation to 0 (depth of root or node)

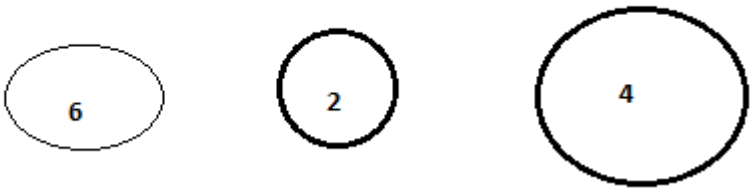

Fig2

The equation of left node is considered and got the value 4 and compares with data 6.If it the nearest value then add it to the tree. Continue the operation like that attain the full tree

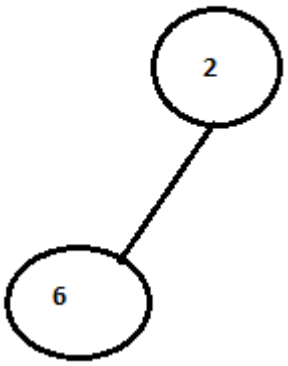

Fig3

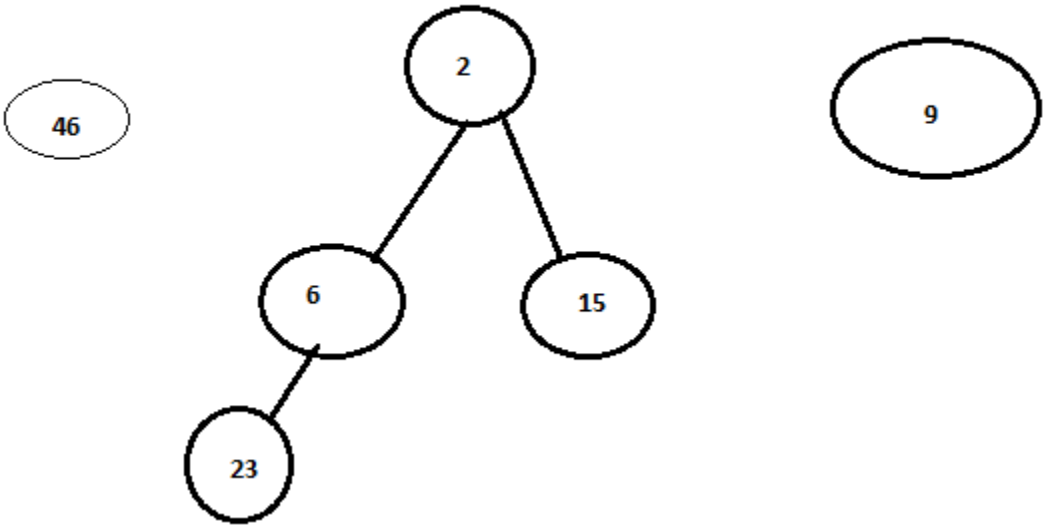

Fig4

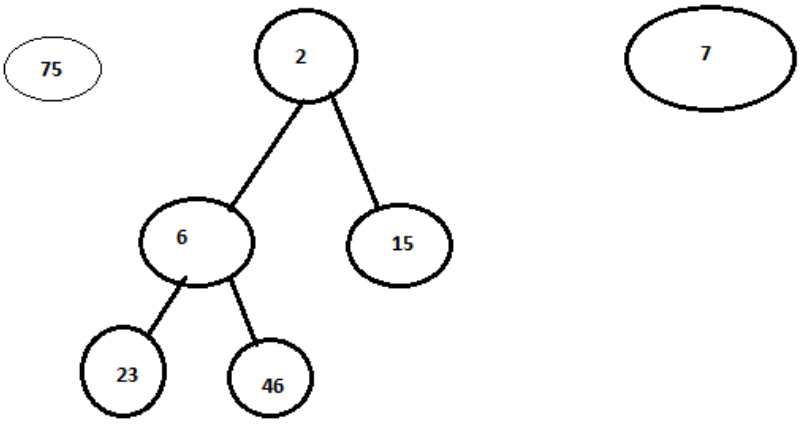

Fig5

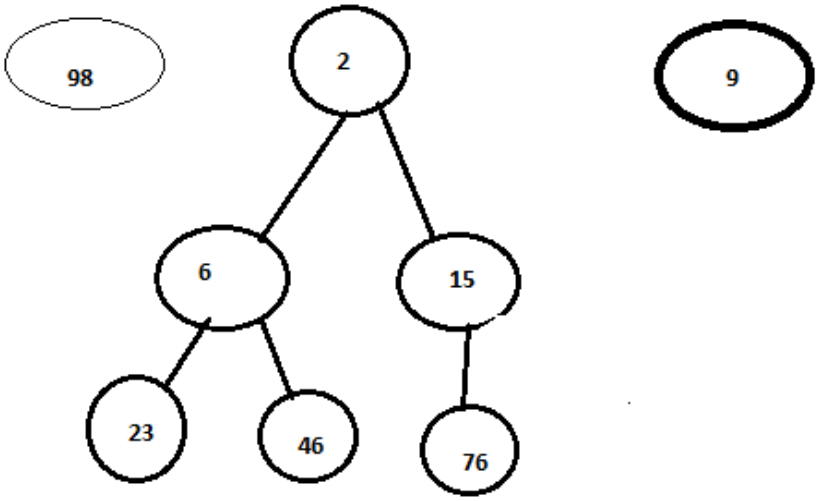

Fig6

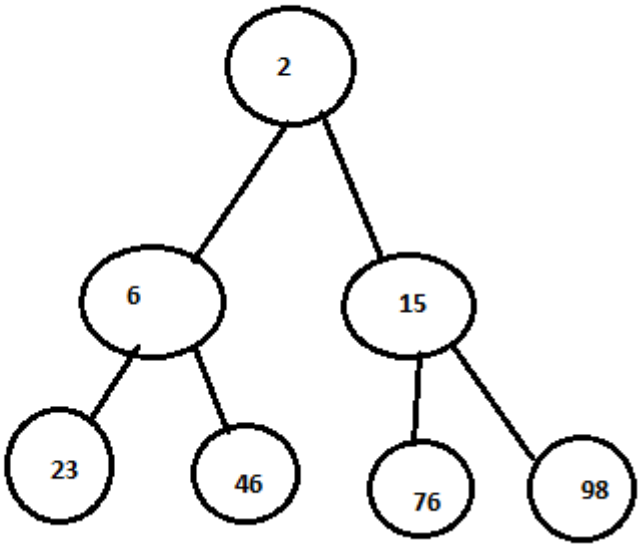

Fig7

IV. CONCLUSION

The proposed ABL tree algorithm provide high performance and security

REFERENCES

[1] Data Structure ,Algorithm and Applications in C++,Satraj Sahni

Mr. Binu.C.T is a post graduate in Computer Science and Engineering. He have 3 Years' Experience in Academic field in different Engineering Colleges. He also have 2.3 Years' experience in software Testing Domain and currently working in Syntrio technologies as Software Test Engineer