ANNALS OF THE FACULTY OF ENGINEERING HUNEDOARA

2005, Tome III, Fascicole 3

MULTIMEDIA EDUCATIONAL SOFTWARE FOR BINARY TREE AND THEIR APPLICATION PRESENT

PĂNOIU Manuela*, MUSCALAGIU Ionel*,
OSACI Mihaela*, IORDAN Anca*

UNIVERSITY POLITEHNICA TIMIŞOARA,
FACULTY ENGINEERING OF HUNEDOARA, ROMANIA

ABSTRACT

The informative society needs important changes in educational programs. The informational techniques needs a reconsideration of the learning process, of the programs, manuals structures, a reconsideration of the methods and organization forms of the didactic activities, taking into account the computer assisted instruction and self instruction. This paper presents a software package, which can be used as educational software.

This paper present a graphical user interface implement in Borland Delphi useful for computer based training. It is made a study of binary tree and their application.

KEYWORDS

binary tree, data structures, educational software

1. INTRODUCTION

Today, it is use the computer in many areas, including in education, from the elementary school to the universities, by following some purposes. Between these purposes it can be cite the necessity to prepare the young people for a future where the computer science is a dominant area. This purpose can be realized by using new didactical teaching methods based on educational software that is programs useful in teaching-learning process [1][2].

By using the computer in didactical activity it is increase the learning productivity: the necessary information is faster and accurate obtained, process and sends, by eliminate unnecessary time delay.

It can be showing to the student in short time some information which is inaccessible otherwise: dynamic diagrams, moving images, sounds. The computer can quickly process all the available information, by a relevant manner present. To exemplify, for many disciplines study – especially experimental disciplines – the computer facilities can be used. Many complex physical and chemical processes are inaccessible to a spatial and temporal scale for the human sense.

The educational software has an important characteristic: it is and can be interactive. Thus, the user can interfere with the program by changing on line some parameters for the studies process.

By using the educational software, the computer can be as well a board, a didactic movie, an oscilloscope, a questionnaire and also other didactical tools which are used in didactic activities.

From desire of improve the instructive educational process, using modern teaching methods, was realized a courseware where the subject "*Binary tree and their applications"* was especially developed.

2. BINARY TREE DATA STRUCTURES

Binary tree is one of the most important data structures in the programming world. The basic definition can be given as follows [3]: a *binary tree is a finite set of elements that is either empty or is partitioned into three disjoint subsets. The first subset contains a single element called root of the tree. The other two subsets are themselves binary trees, called left and right sub trees of the original tree. A left or right sub tree can be empty and each element off the binary tree is called as the node.* The definition itself is recursive in nature. Te follow diagram would make this clearer.

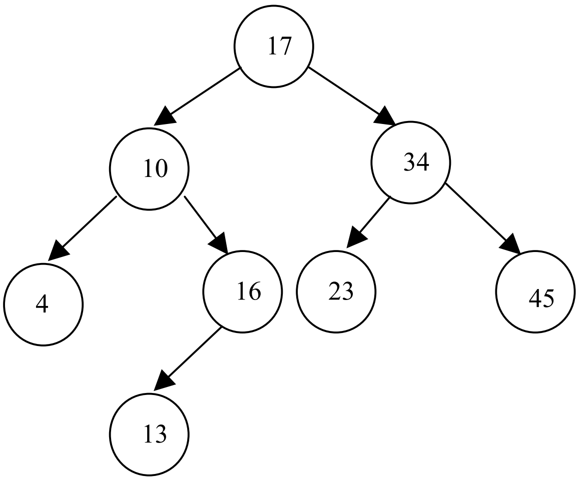


Figure 1. A sorted binary tree

There are countless variations to the basic the binary tree defined and each of these has various applications. For example, the Heap is an implementation of the binary tree. Various low level data structures of the file system and databases would be a heap. Another application of binary trees is searching data in a sorted binary tree.

Implementing a binary tree

In this paper we shall see a very simple binary tree implementation. In terms of implementation, a binary tree is a type of a linked list of nodes. So, a node can be defined:

```
Type  pnode=^node;
        node=record
        info:object_type;
        left,right:pnode;
     end;
```

Here, we have an object called pnode with members Left and Right to point to a left and right sub tree respectively, and info which is type *object_type* (can hold anything). Creating a tree itself, again, is pretty simple. First, we create the root and then create the right and left sub trees. In our example, in visual application, we create a simple tree of char.

We represent the Tree as a object class which exposes several methods to operate on it. Note that the root of the tree can be private and the only way to operate on the tree is through methods exposed by the Tree object class. To create a new tree, we create a new node in the class constructor and mark it as the root.

Adding nodes to the tree

Adding a node to a tree is a bit more involved effort as a node can fit in one and only one place in a tree after a root is in place. We need to start with the root and compare the value of the root with the node's value. If the node value is smaller, then it goes to left sub tree and if it is greater, then it forms a part of the right sub tree. These comparisons happen all the way to the bottom of the tree, where we cannot proceed any further. At this point, one final comparison is made and the node is added to the final node's left or right sub tree. In our implementation, we don't support duplicate nodes. If in the process of adding a node, if its value is equal to node being compared, it is not added to the tree.

Note that in our application, we making comparisons directly on the node value itself. This is under the assumption that all the nodes represent simple char. Ideally, we should have a Key for each node and the comparisons should happen on this field. In terms of implementation, this Key object should implement *Compare* interface so that the tree can be of any type of object (the comparison would be transparent).

Traversing the tree

Traversing a tree involves visiting all the nodes in the tree [5][6][7]. In a traversal a tree each node is visited exactly once. During the visit of a node, all action (display, evaluate an operator, etc) with respect to this element is taken. There are three ways you can achieve this:

- Post Order Traversal
- Pre Order Traversal
- In Order Traversal

Each of these approaches is very useful and has its own applications. By example, we shall consider Inorder traversal. This is a recursive process as explained below:

1. Traverse the left sub tree
2. Visit the root
3. Traverse the right subtree

The significance of this traversal is that we visit all the nodes in a sorted fashion. So, in our case, we shall obtain the sorted list of chars by just traversing in order.

3. APPLICATION PRESENT

The application is implemented in Borland Delphi 6.0, under Microsoft Windows operating system. To implement the tree it was defined an object type which include the following elements: the node structure (a char, the left pointer and right pointer), the constructor and the destructor and all the procedures for creating the tree and for traversing the tree.

The graphical use interface was structured in three parts: a theoretical presentation part, a simulation part and a practical application part.

From the main application window, it can select by a main menu one of these parts. After an option is selected, a new window is opened. First, a theoretical presentation for the selected lesson was present. The user can read this theoretical part, and it can print this or view a simulation, or it can view one of the main applications of the binary trees. The main window is representing in figure 2.

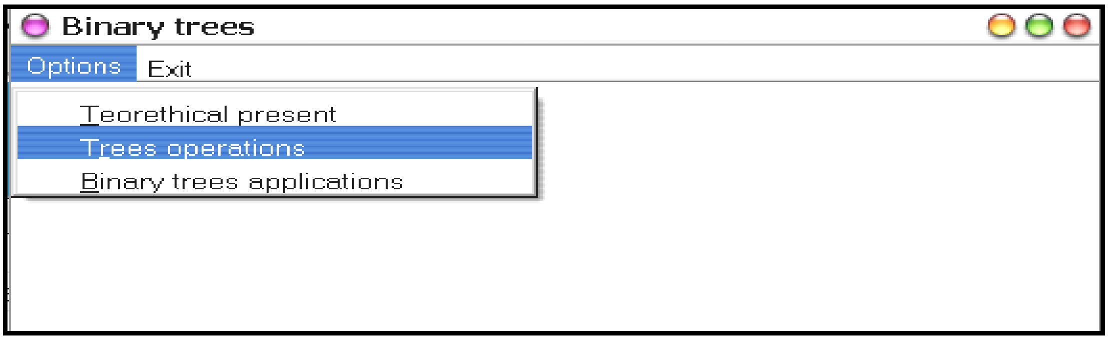
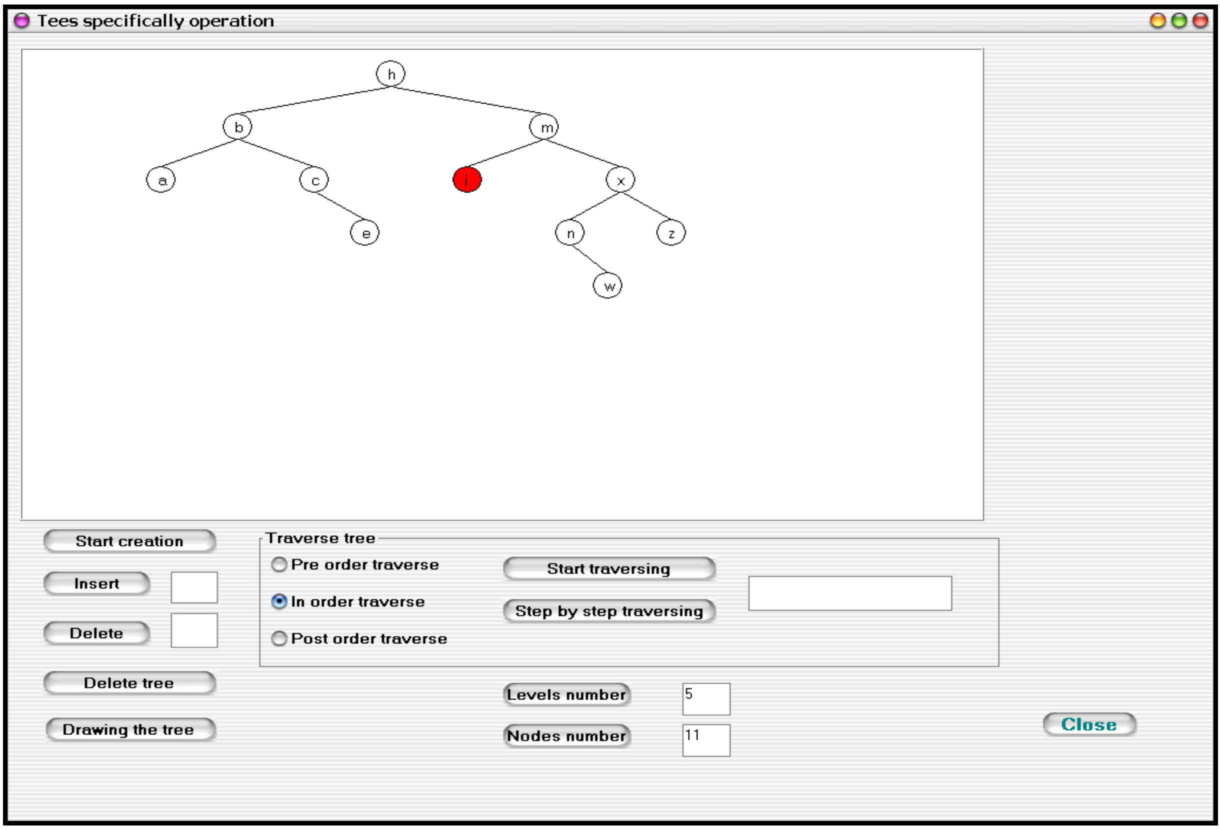


Figure 2. The main application window

One of the most interesting options is the „Trees operation”. This option presents, step by step, all the main operation with a binary tree. This option was developed in another window, present in figure 3.

In this window are presented all the common operation with the nodes of tree. The user has the possibility, in an interactive mode to select and understand these operations. One of the most important characteristic of a courseware (educational software) is the possibility to be interactive.

At the start, there is no node in the tree (it is an empty tree). By click the „Start creating” button the user can add the nodes successively. It is also possible to delete a node from the tree. The node which are inserting or deleted is displayed with red color for more accuracy. The user can observe that after any operation (insert a node or delete a node) the tree remain sorted, which is very useful. In many real situation, when it is works with large set of dates, it is necessary to sort this dates by various criteria. By example, a names list must be alphabetic sorted. Date sorting may be an intermediary step to the searching operations. There are many sorting algorithms, classified by their efficiency and by execution time [5][6][7]. The efficiency of an algorithm is measured by some indicators. Such an indicator is representing by the number of the key comparison, other indicator is the number of element assignation, respectively the number of the elements inversion [5][6][7]. Both two indicators depend by the number of elements, n, which will be sorted.



○ Pre order traverse
● In order traverse
○ Post order traverse

Levels number 5

Nodes number 11

Figure 3. Trees specifically operation

In this case, by using binary trees, all these operations (inversion of two elements, or assignation, or shift element into memory) are not necessary because any operation we made, the tree remain sorted. By using the „Insert” and „Delete” buttons these conclusion are obviously. In the same way it is possible to search a data in a sorted tree. The search algorithm is very fast in this case.

In case that it is necessary, the whole tree can be deleted, using the “Delete tree” button, and the create process can start at the beginning.

The three ways for traverse the tree can be present by using the three radio buttons grouped under the name “Traverse tree”. Each radio button can be use for traversing the tree step by step, or continuously. By step by step traversing, the user must use the “Step by step” button after a node if processed. In this application, when a node is processed, it is show with another color (red) for much clarity.

Another way is to continuously traverse the tree, by using the “start traversing” button. In this case the user can’t interfere with the application until the whole tree is traverse. The process is displayed continuously, and after a time delay it is show the next processed node (with red color).

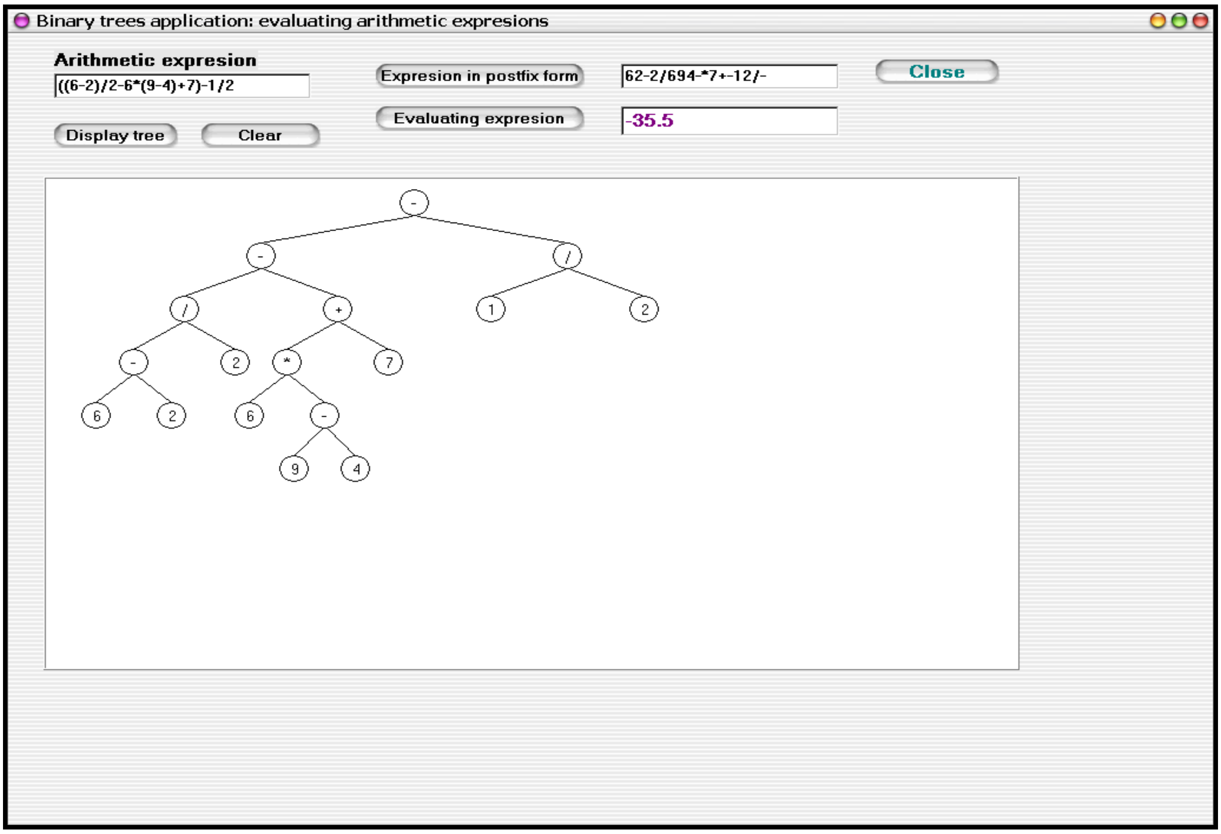
The "number levels" button can be use for tree number levels display, and the "number nodes" can be use for tree number nodes.

It is very well know that the most important application of the sorted binary trees is the arithmetic expression evaluating [5][6][7]. For this application another window was developed.

In this new window, presented in figure 4 the node information it isn't only char but also numerical. So, the terminal nodes are numerical, and the leaf nodes are from the set {'+','-','*','/'} of char which represent the four basic arithmetic operation.

It is also very well know that a source code in any programming language is transformed in several steps, so that finally ill become an executable program. In such a code source an arithmetic expression will be transform in an intermediate form which shall be the object code for this expression. Base on a tree data structure, like in figure 4, for each arithmetic expression it is obtained the postfix form which can be evaluate base on the stack data structure.

In this application we simulate the process. Therefore, the student can tape the arithmetic expression in an edit box (the expression can contain also parenthesis) using the four arithmetic operation and, by using the "Display tree" button the associate expression tree is displayed in a panel. The student also can view the postfix form for the taped expression and evaluate this expression.


Binary trees application: evaluating arithmetic expresions

Arithmetic expresion: ((6-2)/2-6*(9-4)+7)-1/2

Expresion in postfix form: 62-2/694-*7+-12/-

Evaluating expresion: -35.5

Figure 4. Evaluating an arithmetic expression using the associate tree

5. CONCLUSIONS

On this application, authors take into consideration the condition which must accomplish a courseware, being made necessary steps [1][2]. So, in elaboration and utilization of this application must take into consideration next criteria:

- To follow up the curriculum for a specific domain;
- To accomplish some teaching and learning strategy. In this kind of self-instruction and evaluation program it must find basic notions and representation and scanning notions. Animation and graphical modeling must represent the graphical construction way and also scanning of them;
- To exist the possibility to use parameterized variable, in conditions in which users have the possibility to input the variables value;
- To present a method in which the user can be informed about how can use graphical module, i.e. an interaction user-computer exist.
- The presented application accomplishes these criteria, and for this we consider that is a good example of how educational software must be realized.

6. REFERENCES

[1.] Anne Mcdougall, David Squires, Empirical study of a new paradigm for choosing educational software, Computer Education, vol 25, no 3, Elsevier Science, 1995

[2.] GLENN W. ROWE, PETER GREGOR, A computer based learning system for teaching computing: implementation and evaluation, Computer and Education, Elsevier Science, 1999

[3.] Tennenbaum, Langsam & Augenstein, Data Structures Using C and C++, 2e, Prentice Hall Publication, 1996

[4.] BIFFAH HANCILES, VENKY SHANKARAMAN and JOSE MUNOZ, Multiple representation for understanding data structures, Computer Education, vol 29. no. 1, Elsevier Science, 1997

[5.] ATANASIU A., Bazele informaticii, Universitatea din Bucureşti,1989

[6.] KNUTH D. E., Tratat de programarea calculatoarelor. Algoritmi fundamentali, Editura Tehnică, Bucureşti

[7.] KNUTH D. E., Tratat de programarea calculatoarelor. Sortare şi căutare, Editura Tehnică, Bucureşti