Assignment No - 3 (Group B)

Problem statement:

A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.

Pre-requisite

Knowledge of C++ programming

Knowledge of tree data structure

Objective

1. To understand concept of tree data structure

2. To understand concept & features of object oriented programming.

Input

Book name & its number of sections and subsections along with name.

Output

Formation of tree structure for book and its sections

Software and Hardware requirements:-
1. **Operating system:** Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 10,
2. **RAM-** 2GB RAM (4GB preferable)
3. C++ / gcc compiler- / 64 bit Fedora, eclipse IDE

Theory-

Definition:

A tree T is a set of nodes storing elements such that the nodes have a parent-child relationship that satisfies the following

• if T is not empty, T has a special tree called the root that has no parent

• each node v of T different than the root has a unique parent node w; each node with parent w is a child of w

Tree is a widely-used data structure that emulates a tree structure with a set of linked nodes. The tree graphicaly is represented most commonly as on *Fig1*. The circles are the nodes and the edges are the links between them.

Trees are usualy used to store and represent data in some hierarhical order. The data are stored in the nodes, from which the tree is consisted of.The topmost node in a tree is called the root node.

A subtree is a portion of a tree data structure that can be viewed as a complete tree in itself.

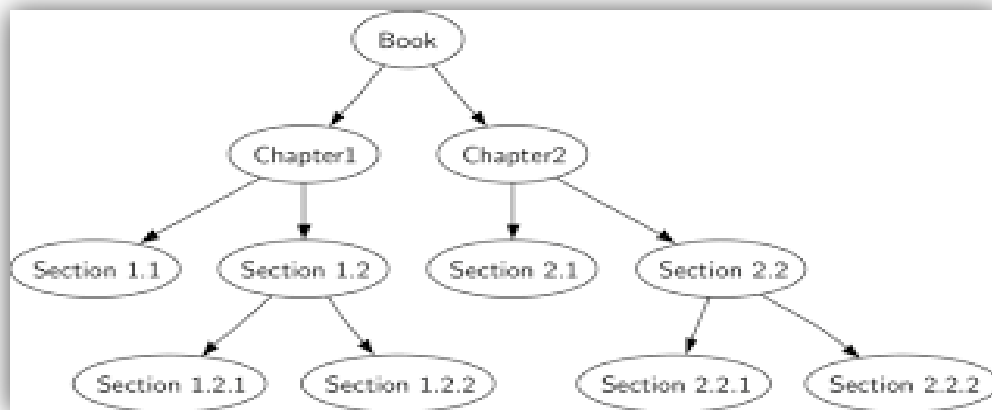For this assignment we are considering the tree as follows.

Fig1. An example of a tree

Representation of a General Tree

We can use either sequential organization or linked organization for representing a tree
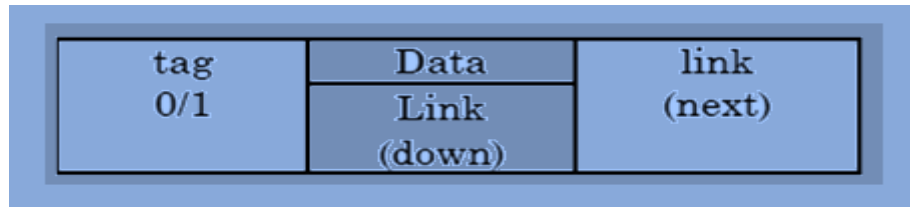
If we wish to use generalized linked list, then a node must have varying number of fields depending upon the number of branches

However, it is simpler to use algorithms for data in which the node size is fixed
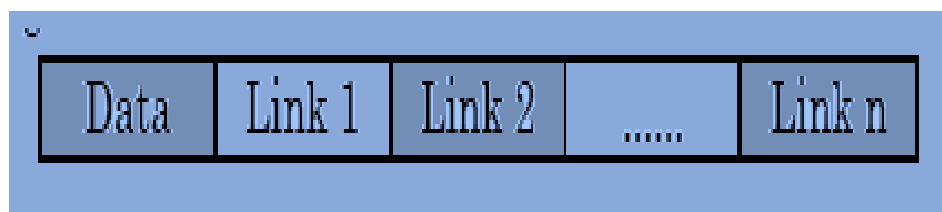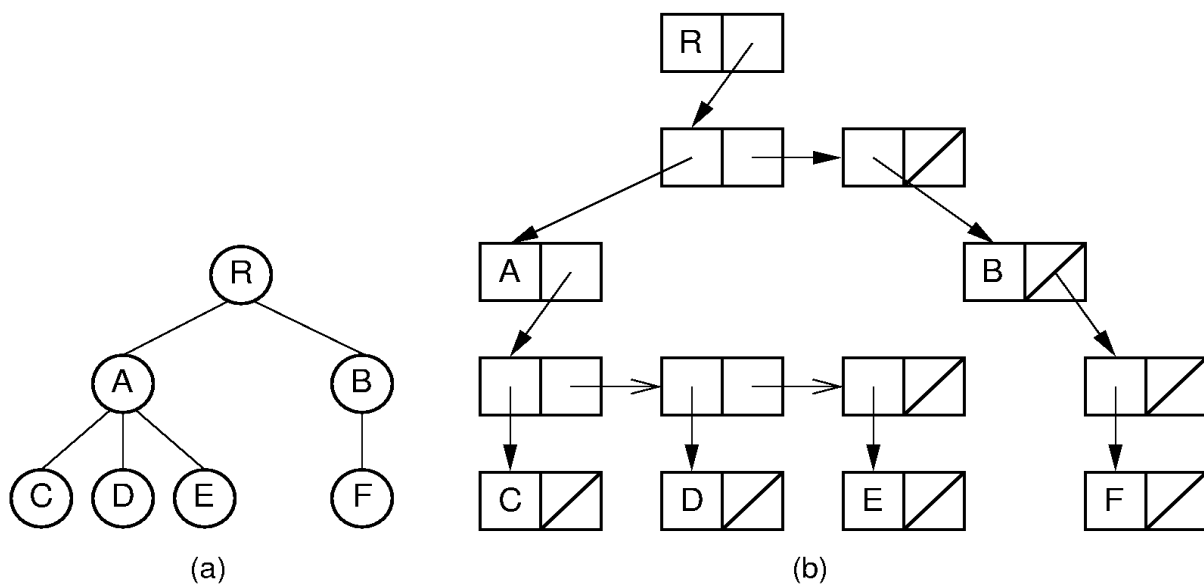
Pseudo code Notations

For a fixed size node, we can use a node with data and pointer fields as in generalized linked list

tag 0/1	Data	link (next)
	Link (down)	

General Tree Example

(a)

(b)

Structure of every node in this tree is:

struct node

{   char name[20];

   node *next;

   node *down;

   int flag;

};

Algorithm:

1. Creating a Tree:

1)Allocate memory for new book node.

2)store following details in the node

   book name and no of chapters

3)for total no of chapters do

    i)Allocate memory for new chapter node and assign its address as downpointer of book node.

    ii)store chapter name and total no of sections in it.

    iii) for total no of sections do

        a) Allocate memory for new section node and assign its address as downpointer of chapter node.

        b) store section name and total no of subsections in it.

        c)for total number of subsections do

            -Allocate memory for new sub section node and assign its address as down pointer of section node.

            -Store details of subsections in the node.

2. Display Tree:

1)If root is not NULL

    i)Display Book Title

    ii)for total no of chapters do

        a)Display chapter name

        b) for total no of sections do

            - Display section name

            - for total number of subsections do

               -Store name of subsections in the node.

Advantages of trees

- Trees reflect structural relationships in the data
- Trees are used to represent hierarchies
- Trees provide an efficient insertion and searching
- Trees are very flexible data, allowing to move subtrees around with minumum effort

Conclusion:

This program gives us the knowledge tree data structure.

Sample Oral Questions :

1. Define the terms :

   General Tree, Binary Tree, internal node, leaf node, siblings, in-degree, out-degree, degree, level, depth/height, path

2. Can we convert a general tree into binary tree? What are the steps to convert a general tree into binary tree? Explain with an example.

3. Explain the binary tree representations.

4. What are the advantages and disadvantages of array representation over linked representation of binary tree?