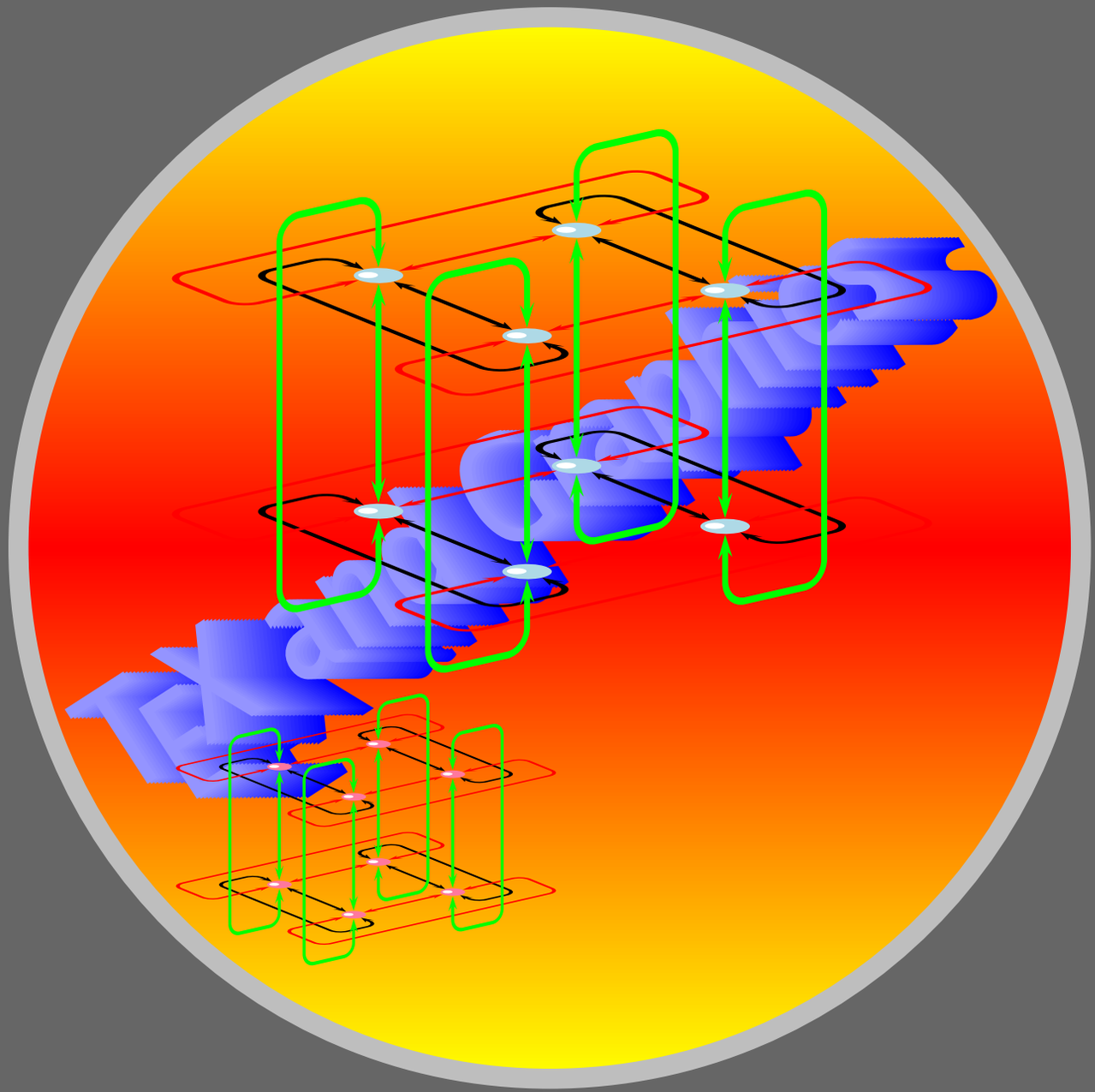


Trees

Online LaTeX Tutorial

Part II – Graphics

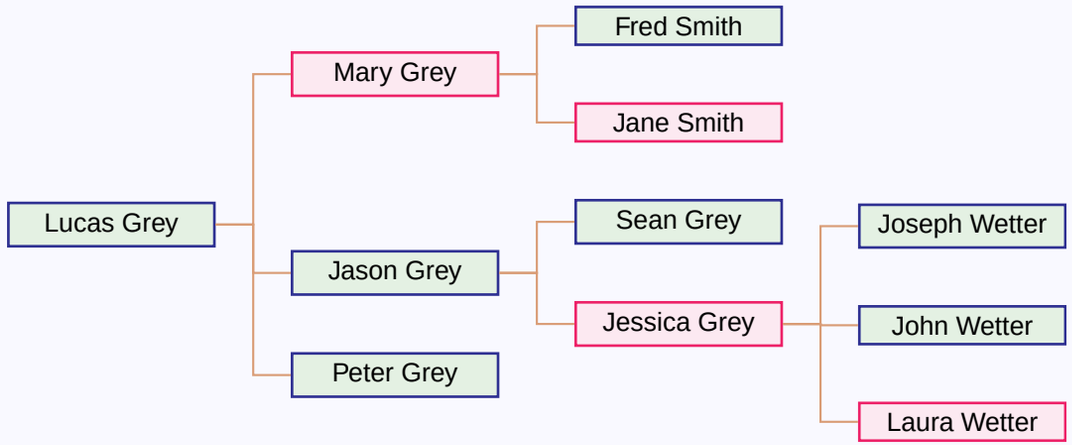
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to
tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11. Trees

This chapter in a sense continues the last chapter, in that we consider nodes and connections again; but here, we discuss special configuration of nodes and connections, which make what is called a *tree*. For example, a *family tree* like this:



or a *factor tree* like this:

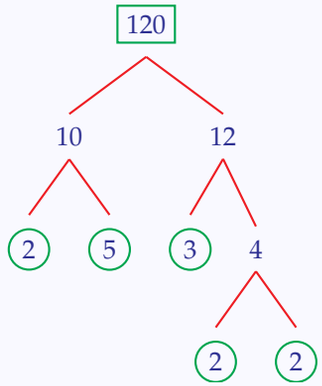
Online LaTeX Tutorial

Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in



With some effort, these could be drawn using the techniques of the last chapter, but the package pst-tree makes it easier. In all the examples below, we have used this package, by declaring `\usepackage{pst-tree}` in the preamble.

Online LaTeX Tutorial

Part II – Graphics

PSTricks

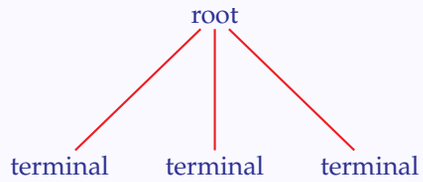
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11.1. Simple trees

The simplest of trees consist of a *root node* and some *terminal nodes* as below:

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red,nodesep=2pt}
  \pstree{\Tr{root}}
    {\Tr{terminal}\Tr{terminal}\Tr{terminal}}
\end{center}
```

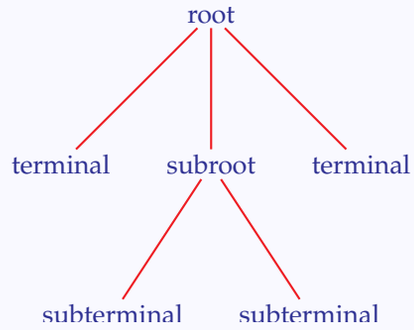


Here, the command `\Tr` is the special version (for trees) of `\rnode` seen in the last chapter. Note the syntax of `\pstree`:

`\pstree{`*root node*`}{`*terminal nodes*`}`

Terminal nodes can also include *sub-trees* as below:

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red,nodesep=2pt}
  \pstree{\Tr{root}}
  {\Tr{terminal}
   \pstree{\Tr{subroot}}
     {\Tr{subterminal}\Tr{subterminal}}
   \Tr{terminal}}
\end{center}
```



Note that this is drawn by replacing the middle terminal of the first example by an entire (sub) tree. Thus the general syntax of the `\pstree` command is

`\pstree{`*root node*`}{`*subtrees and terminal nodes*`}`

We have mentioned that the command `\Tr` creates an `\rnode` in a `\pstree`. Similarly we have `T` versions of the various nodes available in pst-node package. The table below lists these:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

TREE NODE	NORMAL NODE	TREE NODE	NORMAL NODE	TREE NODE	NORMAL NODE
\Tr	\rnode	\Tdia	\dianode	\Tf	\fnode
\TR	\Rnode	\Ttri	\trinode	\Tdot	\dotnode
\Tcircle	\circlenode	\Tc	\cnode	\Tp	\pnode
\Toval	\ovalnode	\TC	\Cnode		

Thus we can get the names of the tree nodes from the normal nodes by stripping the `node` at the end and adding a `T` at the beginning. The syntax for nodes in trees is almost the same, except for a few differences:

- There is an optional argument for setting parameters, even when the original node does not have any such.
- There is no *argument* to specify a name for the node. (We will see later how tree nodes can be given names.)
- There is no coordinate argument for specifying position.
- The reference point in `\Tr` can be set using the `ref` parameter.

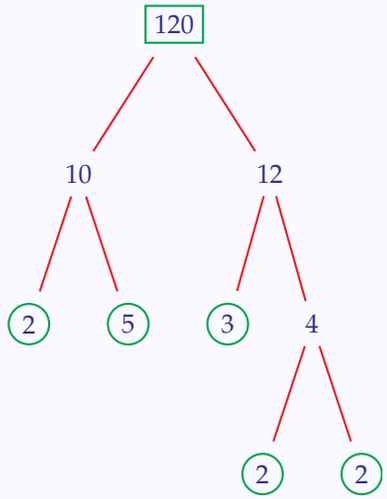
As an example, consider this:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to
tutorialteam@tug.org.in

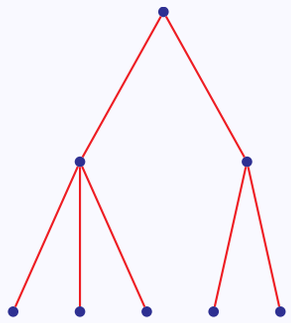
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newcommand{\Tgcircle}{%
    \Tcircle[linecolor=Green]}
  \newcommand{\Tgframe}[1]{%
    \Tr{\psframebox[linecolor=Green]{#1}}}
  \psset{nodesep=5pt,linecolor=Red}
  \color{Blue}
  \pstree{\Tgframe{120}}
    {\pstree{\TR{10}}
       {\Tgcircle{2}\Tgcircle{5}}
     \pstree{\TR{12}}
       {\Tgcircle{3}
        \pstree{\TR{4}}
          {\Tgcircle{2}\Tgcircle{2}}}}
\end{center}
```



This is not quite the same as the one we saw at the beginning of the chapter; we will soon see how we can modify this. As another example, we give a tree with \TC below:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*}
      \pstree{\TBC*}
        {\TBC*\TBC*}}
\end{center}
```



Instead of the \newpsobject command in this example, we can also use the LaTeX \newcommand (as in the previous example) to define our nodes by

```
\newcommand{\TBC}{\TC*[linecolor=Blue,radius=2pt]}
```

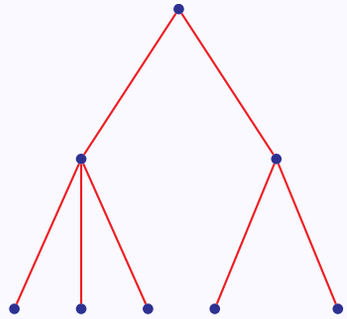

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

and use \TBC instead of \TBC* in the code to get the same output. But the \newpsobject way is more flexible, since we can have both starred and unstarred versions and also change parameter settings at any point.

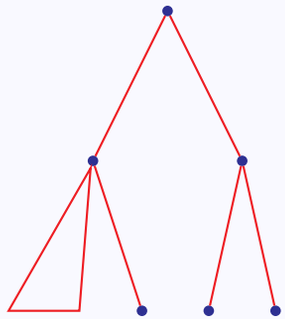
We can also have a *null* node in a tree which is used just as a place-holder. For example, such a node can be used to give a more balanced appearance to the tree in the last example:

```
\begin{center}
   \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*}
      \pstree{\TBC*}
        {\TBC*\Tn\TBC*}}
\end{center}
```



There is another type of tree node called \Tfan for which there is no normal counterpart. It is best seen in action as in the example below:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\Tfan\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*}}
\end{center}
```

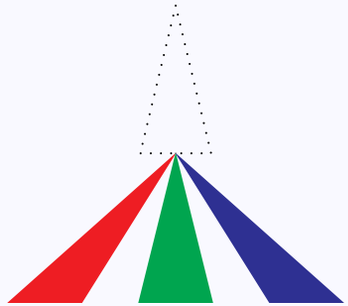


As seen above, \Tfan draws a triangle with its top vertex at the preceding node, with its base horizontal. The width of the base is specified by the fansize parameter, with default value 1 cm. Various parameters for closed graphic objects can be used for \Tfan as in the next example:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \pstree{\Tp}
    {\pstree{\Tfan[linestyle=dotted]}
       {\Tfan*[linecolor=Red]
        \Tfan*[linecolor=Green]
        \Tfan*[linecolor=Blue]}}
\end{center}
```



In this example, we used a point-node `\Tp` as the root since a `\Tfan` cannot be made the root node of an *entire tree*. However, it can be used as the root node of sub-trees as in this example.

Online LaTeX Tutorial
Part II – Graphics
PSTricks

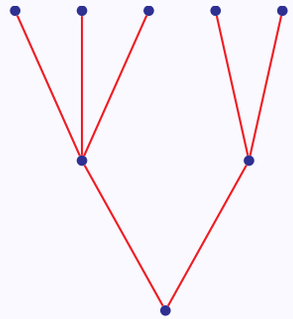
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11.2. Changing direction

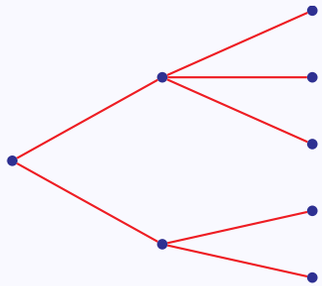
We have seen the the trees that we draw grow downward (unnaturally enough) from the root. This can be changed using the `treemode` parameter. Its possible values are `D`, `U`, `L`, `R`, for trees which grow down (the default), up, left, right. For example, look at one of our earlier example drawn upward:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red,treemode=U}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*\TBC*\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*}}
\end{center}
```



and now the same tree drawn toward the right:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red,treemode=R}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*\TBC*\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*}}
\end{center}
```

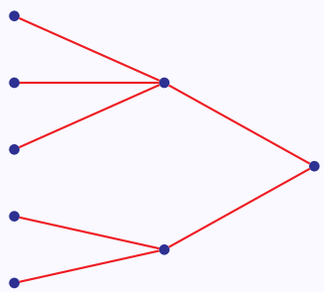


and finally toward the left:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

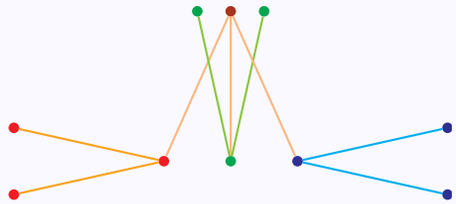
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red,treemode=L}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*\TBC*\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*}}
\end{center}
```



Instead of setting the `treemode` parameter via `\psset`, as we have done, it can be given as an option to the `\pstree` command. This is especially useful when we want to change the direction of only certain subtrees as in the example below:

```
\begin{center}
  \newpsobject{TMC}{TC}{linecolor=Mahogany}
  \newpsobject{TRC}{TC}{linecolor=Red}
  \newpsobject{TGC}{TC}{linecolor=Green}
  \newpsobject{TBC}{TC}{linecolor=Blue}
  \psset{radius=2pt}
  \pstree[linecolor=Apricot]{\TMC*}
    {\pstree[treemode=L,linecolor=YellowOrange]
       {\TRC*}{\TRC*\TRC*}
     \pstree[treemode=U,linecolor=LimeGreen]
       {\TGC*}{\TGC*\TGC*}
     \pstree[treemode=R,linecolor=Cyan]
       {\TBC*}{\TBC*\TBC*}}
\end{center}
```



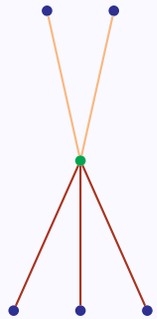
We can also draw two trees of different orientations sharing a common node as below:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

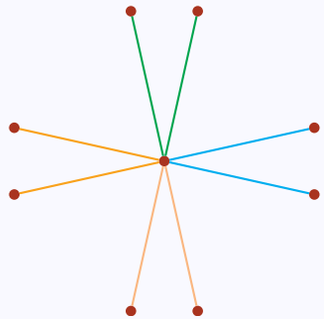
```
1 \begin{center}
2   \newpsobject{TBC}{TC}{linecolor=Blue}
3   \newpsobject{TGC}{TC}{linecolor=Green,radius=2pt}
4   \pstree[linecolor=Apricot,treemode=U]{%
5     \pstree[linecolor=Mahogany]{\TGC*}
6       {\TBC*\TBC*\TBC*}}
7     {\TBC*\TBC*}
8 \end{center}
```



Note that in this code, the tree starting at line 4 has as its root node, the *entire tree* given by lines 5 and 6; and so in the drawing, the mahogany tree and the apricot tree share the same (green) root node.

We can also draw multiple trees sharing a common root node, by nesting `\pstree`'s as below:

```
\begin{center}
 \newpsobject{TMC}{TC}{linecolor=Mahogany,radius=2pt}
  \pstree[linecolor=Apricot]{%
    \pstree[treemode=U,linecolor=Green]{%
      \pstree[treemode=L,linecolor=YellowOrange]{%
        \pstree[treemode=R,linecolor=Cyan]{\TMC*}
          {\TMC*\TMC*}}
        {\TMC*\TMC*}}
      {\TMC*\TMC*}}
    {\TMC*\TMC*}
\end{center}
```



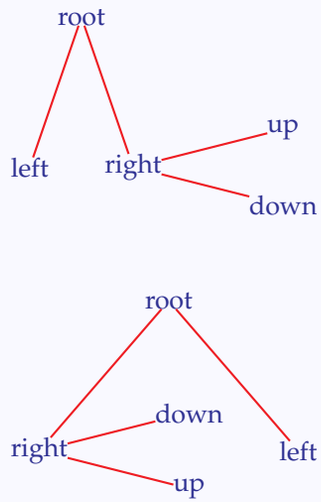
Another parameter for changing the orientation of trees is `treeflip` with values `true` or `false`. The default value is `false`. If set to `true`, it flips the each tree about its root, left to right for vertical trees and top to bottom for horizontal trees:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

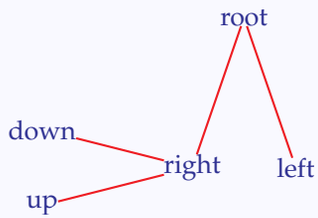
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
    \psset{linecolor=Red}
    \color{Blue}
    \pstree{\Tr{root}}
      {\Tr{left}
       \pstree[treemode=R]{\Tr{right}}
         {\Tr{up}\Tr{down}}}

    \vspace{1cm}

    \psset{linecolor=Red,treeflip=true}
    \color{Blue}
    \pstree{\Tr{root}}
      {\Tr{left}
       \pstree[treemode=R]{\Tr{right}}
         {\Tr{up}\Tr{down}}}
 \end{center}
```



Note that in the above example, the mirror image of the original tree can be obtained by simply changing the `treemode` of the subtree.

```
\begin{center}
  \psset{linecolor=Red,treeflip=true}
  \color{Blue}
  \pstree{\Tr{root}}
    {\Tr{left}
     \pstree[treemode=L]{\Tr{right}}
       {\Tr{up}\Tr{down}}}
\end{center}
```



Online LaTeX Tutorial

Part II – Graphics

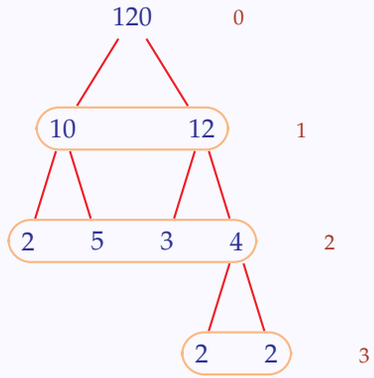
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11.3. Stretching and shrinking

We next see how trees can be made tall or short, thin or thick. First some tree jargon: a tree has different *levels*—the root node of the whole tree is at level 0 and direct descendants of the root are at level 1; if any node at level 1 is the root node of a subtree, then its direct descendants are at level 2 and so on. This is bast shown by a picture:



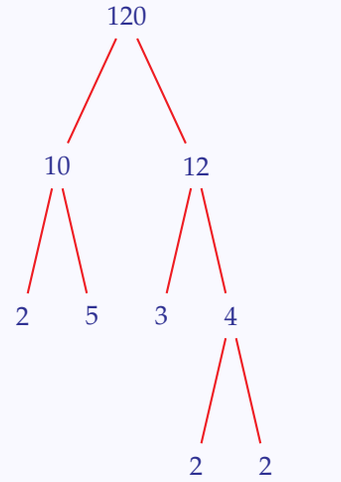
Nodes in the same level, in the order in which they occur, are called *successors*. Thus, for example, in the above picture, 12 is the successor of 10 (in level 1) and 3 is the successor of 5 (in level 2)

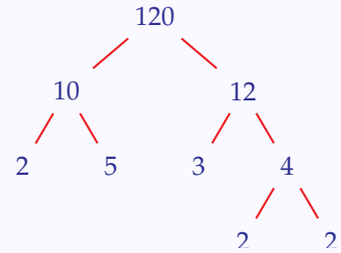
The distance between successors is controlled by the parameter `treesep` (with default value 0.75 cm) and the distance between levels is controlled by the `levelsep` parameter (with default value 2 cm). In the example below, we draw the same tree with default and customized values for `treesep` and `levelsep`

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \psset{nodesep=5pt,%
         linecolor=Red}
  \color{Blue}
  \pstree{\Tr{120}}
    {\pstree{\Tr{10}}
       {\Tr{2}\Tr{5}}
     \pstree{\Tr{12}}
       {\Tr{3}
        \pstree{\Tr{4}}
          {\Tr{2}\Tr{2}}}}

  \vspace{1cm}

  \psset{treesep=1cm,%
         levelsep=1cm,%
         nodesep=5pt,%
         linecolor=Red}
  \color{Blue}
  \pstree{\Tr{120}}
    {\pstree{\Tr{10}}
       {\Tr{2}\Tr{5}}
     \pstree{\Tr{12}}
       {\Tr{3}
        \pstree{\Tr{4}}
          {\Tr{2}\Tr{2}}}}
\end{center}
```





Note that *every* pair of successors cannot be maintained at the same distance, because of the possible presence of subtrees, as these pictures clearly show.

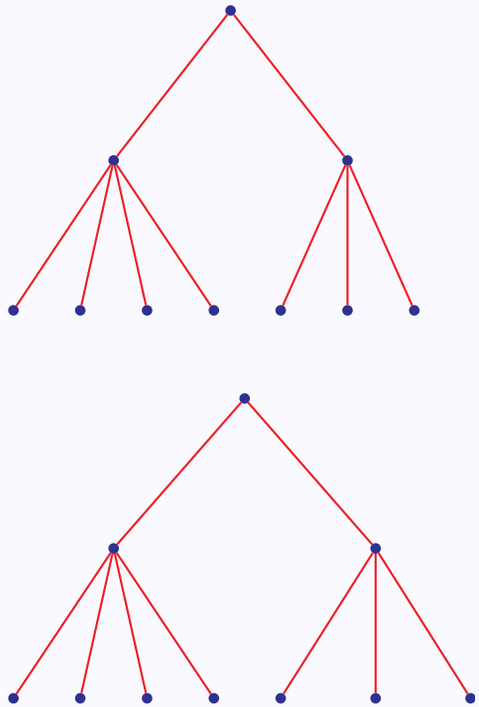
As with other parameters, we can set `treesep` and `levelsep` as options to individual trees, instead using `\psset` to set them globally. See this example, where we give a more balanced look to the tree using appropriate `treesep`. (Note that the earlier trick of introducing an empty node using `\Tn` cannot be used here).

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newpsobject{TBC}{TC}%
              {linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*\TBC*\TBC*\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*\TBC*}}

  \vspace{1cm}

  \newpsobject{TBC}{TC}%
              {linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*\TBC*}
      \pstree[treesep=1.125cm]{\TBC*}
        {\TBC*\TBC*\TBC*}}
\end{center}
```



Here the overall width of the left subtree is $3 \times 0.75 = 2.25$ and by setting half of this length as the `treesep` for right subtree (which has only three terminals), it is also made to have same overall size of the first one.

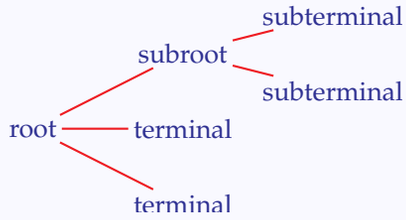
In positioning the successors at each level, separated by `treesep`, the sizes of the nodes are also considered, but in positioning the different levels, they are *not*, by default. This is not much of a problem in vertically growing trees, unless there are very tall nodes. But it is indeed a problem in horizontally growing trees as this example shows:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

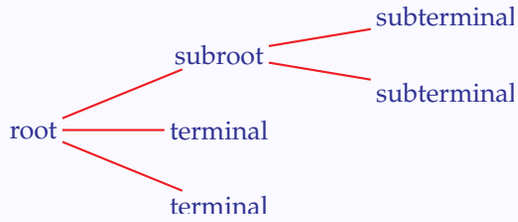
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red,%
         nodesep=2pt}
  \pstree[treemode=R]{\Tr{root}}
    {\pstree{\Tr{subroot}}
       {\Tr{subterminal}{\Tr{subterminal}}}
     \Tr{terminal}\Tr{terminal}}
\end{center}
```



Note that in this picture, though the default value of 2 centimeters is used for `levelsep`, the texts in different (horizontal) levels are not separated by 2 cm. This can be corrected by setting `levelsep=`**length*, in which case, `levelsep` is the specified *length* in addition to the length of the nodes, as shown below:

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red,%
         nodesep=2pt,%
         treemode=R,%
         levelsep=*1.5cm}
  \pstree{\Tr{root}}
    {\pstree{\Tr{subroot}}
       {\Tr{subterminal}{\Tr{subterminal}}}
     \Tr{terminal}\Tr{terminal}}
\end{center}
```



Note that if `levelsep` is set using the optional *, then *two TEX runs are needed to get the positioning right*. Here's a final example on `levelsep` setting:

Online LaTeX Tutorial

Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`

```
\begin{center}
  \scalebox{0.66}{%
    \psset{linecolor=OliveGreen}
    \pstree[levelsep=2cm]{%
      \pstree[levelsep=3.5cm]{%
        \pstree[levelsep=5cm]{%
          \pstree[levelsep=6.5cm]{\Tp}
            {\pstree[levelsep=2cm,%
                     linewidth=1cm,%
                     linecolor=Brown]%
                   {\Tfan*[fansize=3.5cm]}
              {\Tp}}}
          {\Tfan*[fansize=3cm]}}
        {\Tfan*[fansize=2.5cm]}}
      {\Tfan*[fansize=2cm]}}
\end{center}
```

Online LaTeX Tutorial

Part II – Graphics

PSTricks

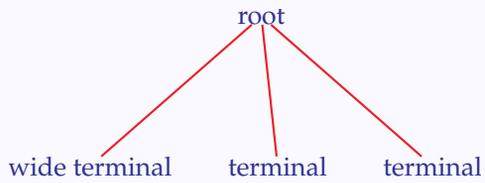
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to
tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

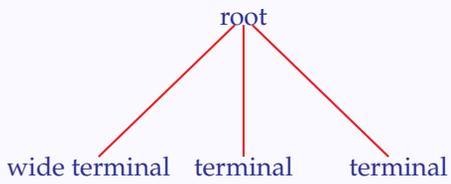
11.4. Fine tuning

The positioning of successors at equal distances may not be to everyone's liking, especially when nodes of different sizes are involved. For example, look at this:

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red}
  \pstree{\Tr{root}}%
    {\Tr{wide terminal}
     \Tr{terminal}
     \Tr{terminal}}
\end{center}
```



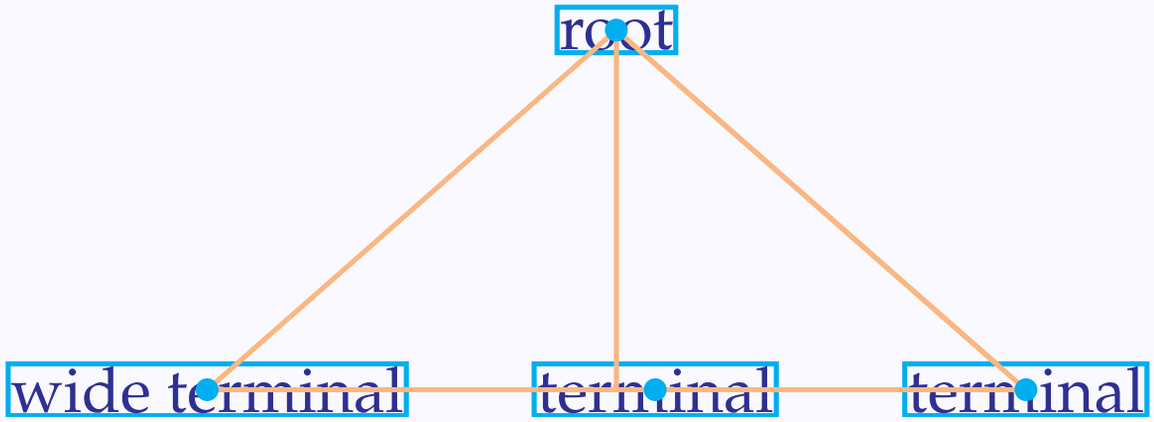
Here the middle line is slanted, because of the longer node at the beginning. Some may prefer to have this line vertical, by bringing the first two nodes at the first level a bit closer, as in this picture:



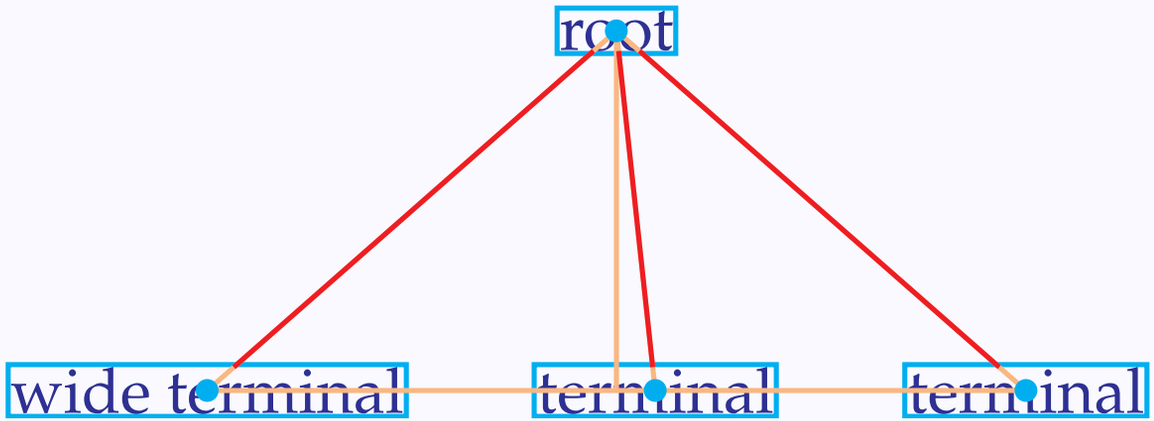
To see how this can be done, let's take a closer look at the positioning of the nodes. With code as above, the three terminal nodes are placed in (invisible) boxes, each of width equal to that of its text, and placed along a horizontal line, with the distance between the edges of adjacent boxes equal to `treesep` The center of the box for the root node is at the top vertex of the isosceles triangle with the line joining the centers of these bottom boxes as base, and height equal to `levelsep`. The magnified picture below shows this:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in



And connectors are drawn joining the mid-points of these boxes.

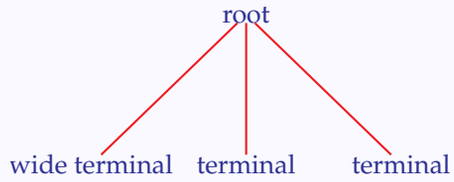


The trick in making the middle line vertical is to force `\pstree` to make boxes of equal width for the terminal nodes. This is achieved by setting the `treenodesize` parameter, as in the example below:

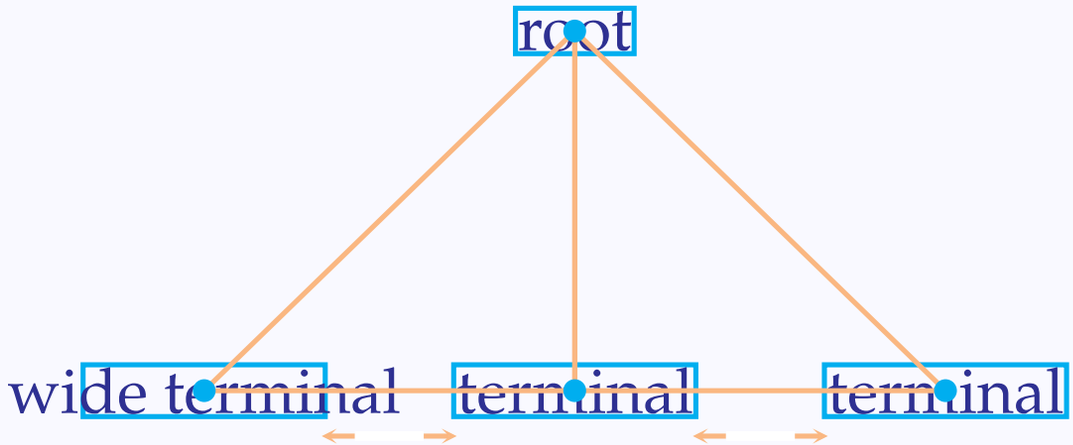
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to
tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\newlength{\tmwd}
\settowidth{\tmwd}{terminal}
\begin{center}
  \color{Blue}
  \psset{linecolor=Red}
  \pstree[treenodesize=0.5\tmwd]{\Tr{root}}%
    {\Tr{wide terminal}
     \Tr{terminal}
     \Tr{terminal}}
\end{center}
```



How does this work? The pair of LaTeX commands `\newlength{\tmwd}` and `\settowidth{\tmwd}{terminal}` define `\tmwd` as the width of the text "terminal". The setting `\treenodesize=0.5\tmwd` puts every terminal node into a box of width `\tmwd` (half of this length on either side of the center of the box) and keeps successor boxes at a distance of `\treesep`. Then the construction described earlier places the `root` node exactly above the middle `terminal`, as shown below:



Note that this setting keeps the distance between the *centers* of the successors equal to `tmwd + treesep`, so that the distance between the *edges* of the last

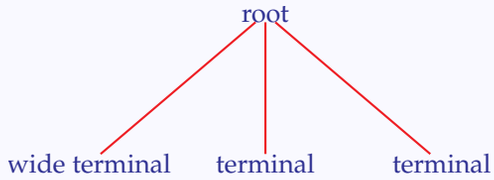
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

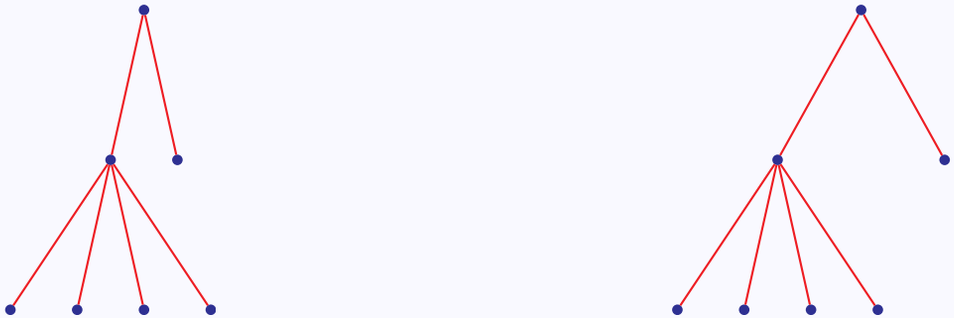
two nodes equal to `treesep`. If one is not interested to keep it thus, any value can be given to `treenodesize`. For example,

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red}
  \pstree[treenodesize=0.8cm]{\Tr{root}}%
    {\Tr{wide terminal}
     \Tr{terminal}
     \Tr{terminal}}
\end{center}
```



In this case, the distance between the centers of successors is $2 \times 0.8 +$ `treesep` which in this case works out to be $1.6 + 0.75 = 2.35$ cm.

There is another instance of adjusting the distance between successors. To illustrate this, look at the two trees shown below:

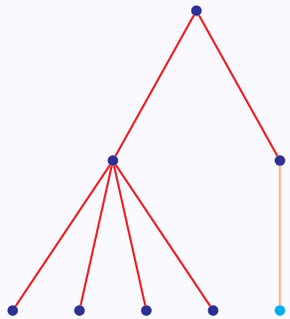


The first tree is what we get by default, with the commands already described. The only change in the second tree is that the successors in the first level are farther apart; in fact in this tree, the *horizontal* distance between the last nodes of the first and second level is also equal to `\treesep`. The picture below illustrates this:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

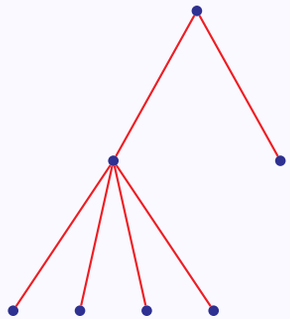
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`



In other words, *all* terminal nodes are equally spaced horizontally. This is achieved by setting the `treefit` parameter to `loose`. (The default value is `tight`) as below:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree[treefit=loose]{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*\TBC*}
      \TBC*}
\end{center}
```



Of course, without the setting `treefit=loose` (or with the setting `treefit=tight`), we get the first (default) tree,

Online LaTeX Tutorial
Part II – Graphics
PSTricks

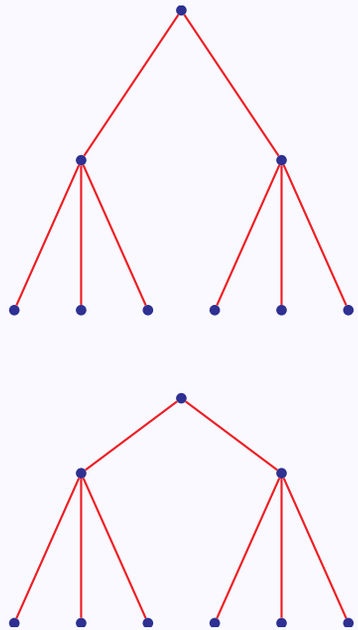
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`

11.5. Local changes

We have seen that parameter changes made with `\psset` affect the entire tree and those made as options to a tree, or a subtree within it, affect all subtrees from that level. However, the four parameters, `treesep`, `levelsep`, `treenodesize` and `treefit` can be changed to affect only the (sub)tree to which they are applied. This is effected by prefixing these parameters with `this`. The example below shows the use of `thislevelsep`.

```
\begin{center}
  \newpsobject{TBC}{TC}%
              {linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*\TBC*\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*\TBC*}}

  \vspace{1cm}

  \newpsobject{TBC}{TC}%
              {linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree[thislevelsep=1cm]{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*}
      \pstree{\TBC*}
        {\TBC*\TBC*\TBC*}}
\end{center}
```

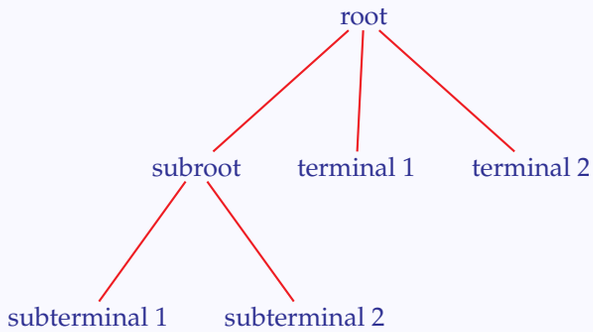


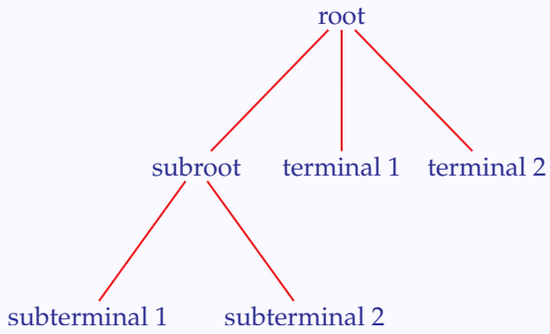
And here's one of our earlier examples modfied (to have a vertical middle line) using `thistreenodesize`

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \color{Blue}
  \psset{linecolor=Red,%
         nodesep=2pt}
  \pstree{\Tr{root}}
    {\pstree{\Tr{subroot}}
       {\Tr{subterminal 1}
        \Tr{subterminal 2}}
     \Tr{terminal 1}
     \Tr{terminal 2}}

  \vspace{1cm}

  \newlength{\srwd}
  \settowidth{\srwd}{subroot}

  \color{Blue}
  \psset{linecolor=Red,%
         nodesep=2pt}
  \pstree[thistreenodesize=0.5\srwd]%
         {\Tr{root}}
    {\pstree{\Tr{subroot}}
       {\Tr{subterminal 1}
        \Tr{subterminal 2}}
     \Tr{terminal 1}
     \Tr{terminal 2}}
\end{center}
```





(What will happen, if we use `treenodesize` instead of `thistreenosesize` in this example? Try it!)

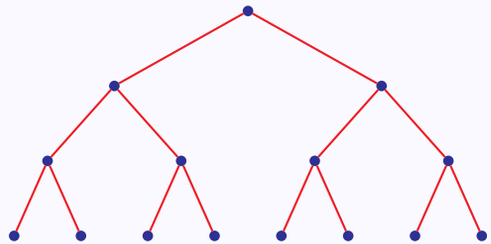
Sometimes we want to make a change of parameter such as `levelsep` uniformly for all subtrees in a particular level. We can of course set `thislevelsep` for *each* of these subtrees; but there is an easier way, by defining the command `\pstreehook`. (Remember the command `\pscolhook` in the `psmatrix` environment, discussed in the last chapter?) Look at this example:

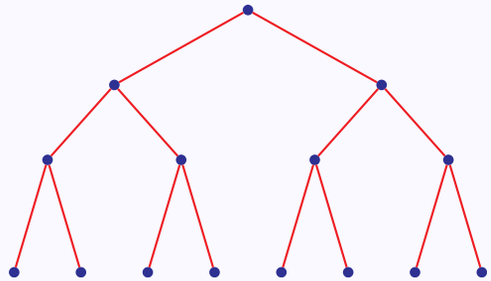
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newpsobject{TBC}{TC}%
             {linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree[levelsep=1cm]{\TBC*}
     {\pstree{\TBC*}
        {\pstree{\TBC*}
           {\TBC*\TBC*}
         \pstree{\TBC*}
           {\TBC*\TBC*}}
      \pstree{\TBC*}
        {\pstree{\TBC*}
           {\TBC*\TBC*}
         \pstree{\TBC*}
           {\TBC*\TBC*}}}


    \vspace{1cm}


   \def\pstreehookiii{%
     \psset{thislevelsep=1.5cm}}
  \newpsobject{TBC}{TC}%
             {linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree[levelsep=1cm]{\TBC*}
     {\pstree{\TBC*}
        {\pstree{\TBC*}
           {\TBC*\TBC*}
         \pstree{\TBC*}
           {\TBC*\TBC*}}
      \pstree{\TBC*}
        {\pstree{\TBC*}
           {\TBC*\TBC*}
         \pstree{\TBC*}
           {\TBC*\TBC*}}}
\end{center}
```





Online LaTeX Tutorial

Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

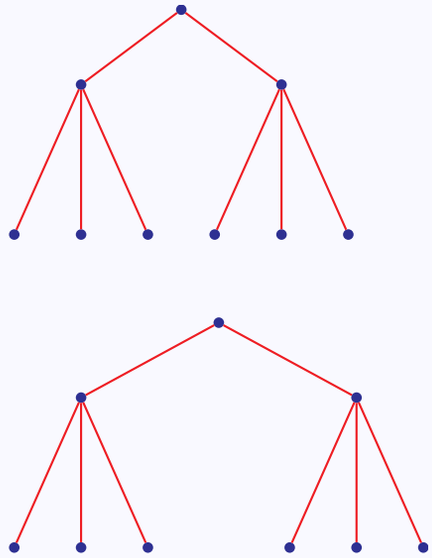
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA

http://www.tug.org.in

Here the line `\def\pstreehookiii{\psset{thislevelsep=1.5cm}}` in the code executes the command `\psset{thislevelsep=1.5cm}` at the third level. (The `iii` at the end of `\pstreehokiii` stands for the *third* level)

There are also a couple of *commands* to make local changes. The command `\tspace{`*length*`}` given between any pair of successors, increases the distance between them by the specified *length*, as in this example:

```
\begin{center}
  \newpsobject{TBC}{TC}%
              {linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree[thislevelsep=1cm]{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*}
      \pstree{\TBC*}
        {\TBC*\TBC*\TBC*}}

  \vspace{1cm}

  \newpsobject{TBC}{TC}%
              {linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \pstree[thislevelsep=1cm]{\TBC*}
     {\pstree{\TBC*}
        {\TBC*\TBC*\TBC*}
      \tspace{1cm}
      \pstree{\TBC*}
        {\TBC*\TBC*\TBC*}}
\end{center}
```



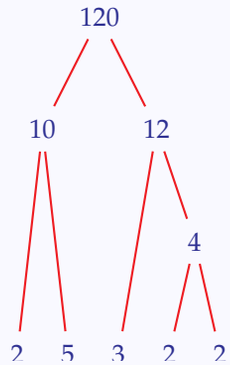
The command `\skiplevel` can be used to push a set of successors down a level. The example below illustrates this with one of our earlier examples:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \psset{nodesep=5pt,%
         treesep=0.5cm,%
         levelsep=1.5cm,%
         linecolor=Red}
  \color{Blue}
  \pstree{\Tr{120}}
    {\pstree{\Tr{10}}
       {\skiplevel{\Tr{2}\Tr{5}}}
     \pstree{\Tr{12}}
       {\skiplevel{\Tr{3}}
        \pstree{\Tr{4}}
          {\Tr{2}\Tr{2}}}}
\end{center}
```



Online LaTeX Tutorial

Part II – Graphics

PSTricks

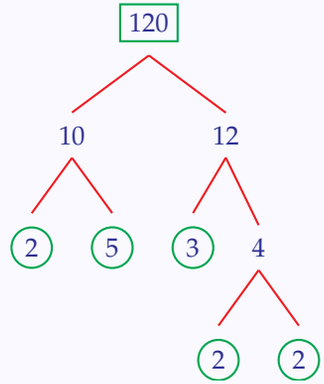
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to
tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11.6. Different edges

You may have noticed that \pstree draws connectors between appropriate nodes without explicit mention in the code. The connectors are drawn using the (internal) command \psedge and by default it is set to \ncline. We can redefine this to be any of the various types of connectors discussed in the last chapter. For example, hers is the code for the factor tree given at the beginning of this chapter, where we use \ncdisg as the connectors:

```
\begin{center}
  \renewcommand{\psedge}{%
    \ncdiag[arm=0,angleA=270,angleB=90]}
  \newcommand{\Tgcircle}{%
    \Tcircle[linecolor=Green]}
  \newcommand{\Tgframe}[1]{%
    \Tr{\psframebox[linecolor=Green]{#1}}}
  \psset{nodesep=5pt,%
         treesep=0.5cm,%
         levelsep=1.5cm,%
         linecolor=Red}
  \color{Blue}
  \pstree{\Tgframe{120}}
    {\pstree{\Tr{10}}
       {\Tgcircle{2}\Tgcircle{5}}
     \pstree{\Tr{12}}
       {\Tgcircle{3}
        \pstree{\Tr{4}}
          {\Tgcircle{2}\Tgcircle{2}}}}
\end{center}
```



The family tree given at the beginning of the chapter is drawn using \ncangle. The code for it is given below:

```
\begin{center}
  \newcommand{\TF}[1]{%
```


E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
    \Tr{\psframebox[fillstyle=solid,%
                    fillcolor=PalePink,%
                    linewidth=1pt,%
                    linecolor=WildStrawberry]{%
                    \makebox[2.5cm][c]{%
                      \sffamily  #1}}}}
  \newcommand{\TM}[1]{%
    \Tr{\psframebox[fillstyle=solid,%
                    fillcolor=PaleBlue,%
                    linewidth=1pt,%
                    linecolor=Blue]{%
                   \makebox[2.5cm][c]{%
                    \sffamily #1}}}}
  \renewcommand{\psedge}{\ncangle[arm=0.5cm,angleB=180]}
  \psset{linecolor=Tan,treemode=R,levelsep=*1cm}
  \pstree{\TM{Lucas Grey}}
    {\pstree{\TF{Mary Grey}}
       {\TM{Fred Smith}\TF{Jane Smith}}
     \pstree{\TM{Jason Grey}}
       {\TM{Sean Grey}
        \pstree{\TF{Jessica Grey}}
          {\TM{Joseph Wetter}\TM{John Wetter}\TF{Laura Wetter}}}
     \TM{Peter Grey}}
\end{center}
```

This could be made more ornate by defining \psedge to be \ncarc and making some cosmetic changes:

```
\begin{center}
  \newcommand{\zapf}{\fontfamily{pzc}\fontseries{m}\itshape}
  \newcommand{\TF}[1]{%
    \Toval[fillstyle=solid,fillcolor=PalePink,%
           linecolor=WildStrawberry,linewidth=1pt]{%
             \makebox[2cm][c]{\zapf\color{Red} #1}}}
  \newcommand{\TM}[1]{%
    \Toval[fillstyle=solid,fillcolor=PaleBlue,%
           linecolor=Blue,linewidth=1pt]{%
             \makebox[2cm][c]{\zapf\color{Red}xs #1}}}
```


Online LaTeX Tutorial

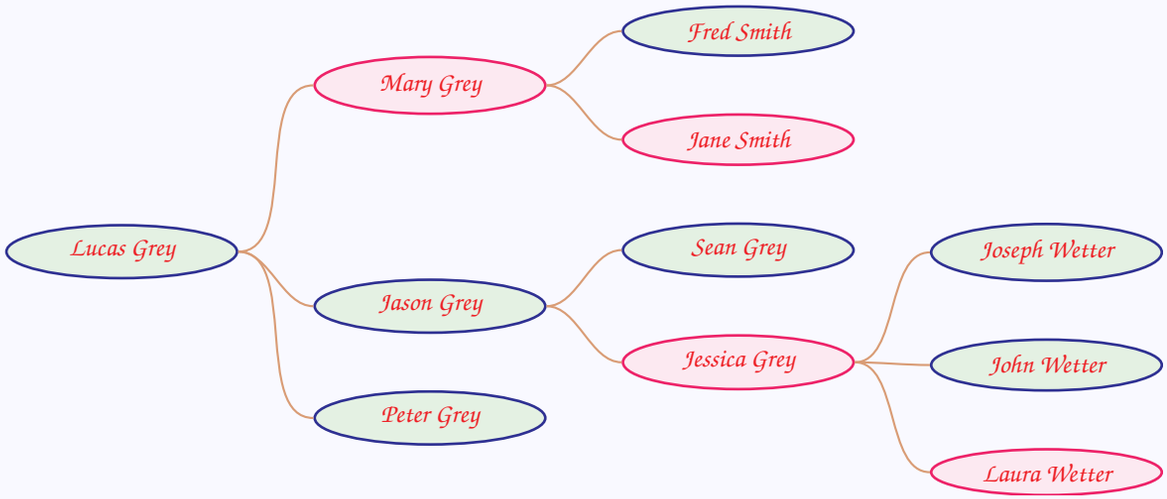
Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
  \renewcommand{\psedge}{\nccurve[angleB=180]}
  \psset{linecolor=Tan,treemode=R,levelsep=*1cm}
  \pstree{\TM{Lucas Grey}}
    {\pstree{\TF{Mary Grey}}
       {\TM{Fred Smith}\TF{Jane Smith}}
     \pstree{\TM{Jason Grey}}
       {\TM{Sean Grey}
        \pstree{\TF{Jessica Grey}}
          {\TM{Joseph Wetter}\TM{John Wetter}\TF{Laura Wetter}}}
     \TM{Peter Grey}}
\end{center}
```



The command \psedge actually has two arguments, the names of the initial and terminal nodes which it connects. So, in redefining \psedge we can have two TEX parameters representing these two nodea. Look at this example:

Online LATEX Tutorial

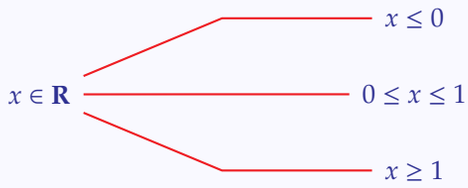
Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

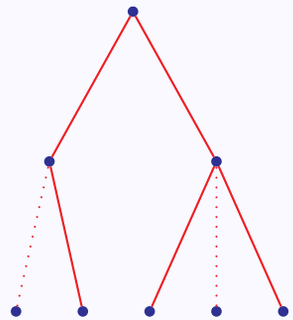
```
\begin{center}
  \renewcommand{\psedge}[2]{%
    \ncdiagg[angle=180,arm=2cm]#2#1}
  \begin{math}
    \color{Blue}
    \psset{linecolor=Red, nodesep=5pt,%
           treemode=R,levelsep=5cm}
    \pstree{\Tr{x\in\mathbf{R}}}
      {\Tr{x\le 0}\Tr{0\le x\le 1}\Tr{x\ge 1}}
  \end{math}
\end{center}
```



Here in the (re)definition of `\psedge`, the final `#2#1` causes a reversal of the usual initial and terminal nodes, so that all connectors are drawn *to* the root node, instead of *from* the node as is usual.

Another way of changing edges is to *define* our own customized edge (rather than *redefine* `\psedge`) and then use the *parameter* `edge` to use our edge, wherever needed. Look at this example

```
\begin{center}
  \newcommand{\dotedge}{%
    \ncline[linestyle=dotted]}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*[edge=\dotedge]\TBC*}
     \pstree{\TBC*}
       {\TBC*\TBC*[edge=\dotedge]\TBC*}}
\end{center}
```

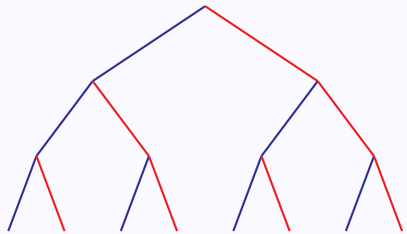


This method of changing edges is especially useful, when we want to switch between different edges frequently, as in the example below:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

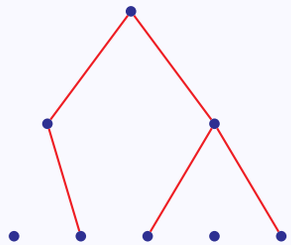
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newcommand{\bedge}{\ncline[linecolor=Blue]}
  \newcommand{\redge}{\ncline[linecolor=Red]}
  \newcommand{\Tpb}{\Tp[edge=\bedge]}
  \newcommand{\Tpr}{\Tp[edge=\redge]}
  \psset{levelsep=1cm}
  \pstree{\Tp}
    {\pstree{\Tpb}
       {\pstree{\Tpb}
          {\Tpb\Tpr}
        \pstree{\Tpr}
           {\Tpb\Tpr}}
     \pstree{\Tpr}
       {\pstree{\Tpb}
          {\Tpb\Tpr}
        \pstree{\Tpr}
          {\Tpb\Tpr}}}
\end{center}
```



We can also set edge=none at any point to suppress a node connection, as in the example below:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red,levelsep=1.5cm}
  \pstree{\TBC*}
    {\pstree{\TBC*}
       {\TBC*[edge=none]\TBC*}
     \pstree{\TBC*}
        {\TBC*\TBC*[edge=none]\TBC*}}
\end{center}
```



As another application of the setting edge=none, we beautify one of our earlier examples as below:

Online LaTeX Tutorial

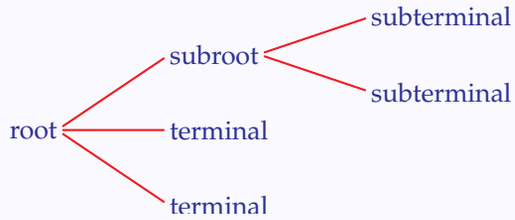
Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
   \newpsobject{Trr}{Tr}{ref=r}
   \newpsobject{Trl}{Tr}{ref=l}
   \color{Blue}
   \psset{linecolor=Red,nodesep=2pt,%
          levelsep=*1.5cm,treemode=R,}
   \pstree{\Trr{root}}
     {\pstree[thislevelsep=*0cm]
             {\Trl{subroot}}
        {\pstree{\Trr[edge=none]{}}
           {\Trl{subterminal}
            \Trl{subterminal}}}
      \Trl{terminal}\Trl{terminal}}
\end{center}
```



Note especially the trick of introducing an *empty tree* in line 10 of the above code. This is to make the connectors *from* this node to start at the right of the text `subroot`. The reference point of this node has been set left of this text by the command `\Trl` in line 9, to make the connector *to* this node end at the left of this text.

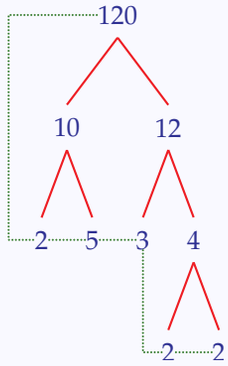
In a tree, we can also connect *any* pair of nodes ourselves.. For this, we have to give name to the nodes to be connected. This can be done using the `name` parameter, as shown in the example below:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`

```
\begin{center}
  \renewcommand{\psedge}{%
    \ncdiag[arm=0,angleA=270,angleB=90]}
  \psset{linecolor=Red,nodesep=5pt,%
         treesep=0.5cm,levelsep=1.5cm}
  \color{Blue}
  \pstree{\Tr[name=n]{120}}
    {\pstree{\Tr{10}}
       {\Tr[name=f1]{2}\Tr[name=f2]{5}}
     \pstree{\Tr{12}}
       {\Tr[name=f3]{3}
        \pstree{\Tr{4}}
          {\Tr[name=f4]{2}\Tr[name=f5]{2}}}}
  \psset{linecolor=OliveGreen,linestyle=dotted,%
         dotsep=0.5pt,nodesep=0pt}
  \ncline{f1}{f2}\ncline{f2}{f3}
  \ncangle[angleA=270,angleB=180,arm=1pt]{f3}{f4}
  \ncline{f4}{f5}\ncbar[angle=180]{n}{f1}
\end{center}
```



Online LaTeX Tutorial

Part II – Graphics

PSTricks

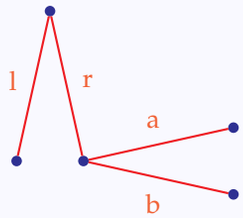
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

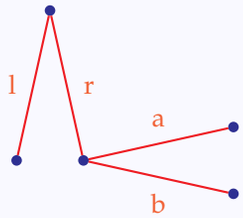
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11.7. Labeling edges

We next see how nodes and edges in a tree can be labeled. We first take up edges. As in the case of node connections in a general graph, edges in trees can also be labeled either with the two \n*put commands \naput and \nbput or the four \t*put commands \taput, \tbput, \tlput and \trput. (See the last chapter for details.)

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{RedOrange}
  \pstree{\TBC*}
  {\TBC*\nbput{$e_1$}
   \pstree[treemode=R]{\TBC*\naput{$e_2$}}
     {\TBC*\naput{$e_3$}\TBC*\nbput{$e_4$}}}

  \vspace{1cm}

  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{RedOrange}
  \pstree{\TBC*}
    {\TBC*\tlput{l}
     \pstree[treemode=R]{\TBC*\trput{r}}
       {\TBC*\taput{a}\TBC*\tbput{b}}}
\end{center}
```





The parameters npos, nrot and tpos can be used in trees also. It must must be noted that within trees, the tpos parameter measures the distance between the nodes connected, whatever be the orientation of the tree, unlike its usual behavior, as explained in the last chapter.

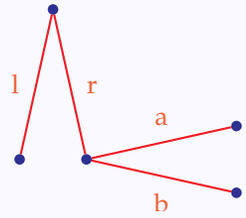
Also, we can use shorter commands ^ and _ for the commands \taput and \tbput in horizontal trees and for \tlput and \trput in vertical trees. .

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

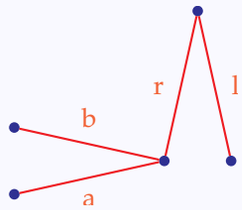
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
   \psset{linecolor=Red}
   \color{RedOrange}
   \pstree{\TBC*}
     {\TBC*^{l}
      \pstree[treemode=R]{\TBC*_{r}}
        {\TBC*^{a}\TBC*_{b}}}
\end{center}
```



Note that the meanings of ^ and _ are reversed if `treeflip=true` is in effect

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red,treeflip=true}
  \color{RedOrange}
  \pstree{\TBC*}
    {\TBC*^{l}
     \pstree[treemode=L]{\TBC*_{r}}
       {\TBC*^{a}\TBC*_{b}}}
\end{center}
```



These short-cuts for labeling edges work because the parameter `shortput` is set to the value `tab` in pst-tree by default. This is a setting available only for trees. (See the last chapter for the values of this parameter which are generally available.) Moreover, these characters can be changed by the command `\MakeShortTab{`*char1*`}{`*char2*`}`, where *char1* and *char2* are any two characters of our choice.

Online LaTeX Tutorial
Part II – Graphics
PSTricks

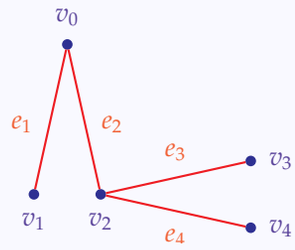
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`

11.8. Labeling nodes

We can use the command \nput to label the nodes, as in the general case discussed in the last chapter. Recall that the syntax of this command is \nput[*parameters*]{*dirangle*}{*name*}{*stuff*} so that to use it, we must have a *name* for the node. In a \pstree, we need not name the nodes ourselves, but can use the name used internally by \pstree. Within a tree, whenever a new node is formed, a connector is also drawn, connecting it with its *predecessor* (if there is one). This is implemented by naming the current node by the command \pssucc and the predecessor by the command \pspred. So, we can label any node in a tree using \nput with \pssucc as the name of the node. Look at this example:

```
\begin{center}
  \newcommand{\TBC}{\TC*[linecolor=Blue,radius=2pt]}
  \psset{linecolor=Red}
  \pstree{\TBC\nput{u}{\pssucc}{\color{Violet} $v_0$}}
    {\TBC\nput{d}{\pssucc}{\color{Violet} $v_1$}
         \tlput{\color{RedOrange} $e_1$}
     \pstree[treemode=R]{\TBC
       \nput{d}{\pssucc}{\color{Violet} $v_2$}
       \trput{\color{RedOrange} $e_2$}}
       {\TBC\nput{r}{\pssucc}{\color{Violet} $v_3$}
            \taput{\color{RedOrange} $e_3$}
        \TBC\nput{r}{\pssucc}{\color{Violet} $v_4$}
             \tbput{\color{RedOrange} $e_4$}}}
\end{center}
```

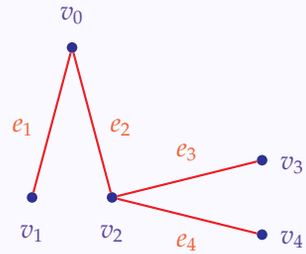


There is a short-cut to labeling nodes also, using the ~ symbol. Thus the last example can also be coded as below:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

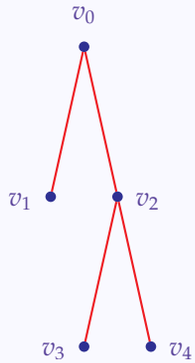
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
   \newcommand{\TBC}{\TC*[linecolor=Blue,radius=2pt]}
   \psset{linecolor=Red}
   \pstree{\TBC~[tnpos=a]{\color{Violet} $v_0$}}
     {\TBC~{\color{Violet} $v_1$}
        \tlput{\color{RedOrange} $e_1$}
      \pstree[treemode=R]{\TBC~{\color{Violet} $v_2$}
        \trput{\color{RedOrange} $e_2$}}
        {\TBC~{\color{Violet} $v_3$}
           \taput{\color{RedOrange} $e_3$}
         \TBC~{\color{Violet} $v_4$}
           \tbput{\color{RedOrange} $e_4$}}}
 \end{center}
```



Note the parameter setting `tnpos=a` in the specification of the label `$v_0$` in this example. This is because, when nodes are labeled using `~`, the labels are positioned automatically and for the root node, the default position of the label is *below* the node. We set it above the mode by the parameter setting `tnpos=a`. The other possible values of this parameter are `b`, `l` and `r` with obvious meanings. The example below illustrates this:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
   \color{Violet}
   \psset{linecolor=Red}
   \pstree{\TBC*~[tnpos=a]{$v_0$}}
     {\TBC*~[tnpos=l]{$v_1$}
      \pstree{\TBC*~[tnpos=r]{$v_2$}}
        {\TBC*~[tnpos=l]{$v_3$}
         \TBC*~[tnpos=r]{$v_4$}}}
 \end{center}
```



A useful feature of the short-cut form of specifying labels is that it reserves space for the labels, unlike the generic `\nput`. (We will discuss this in a little

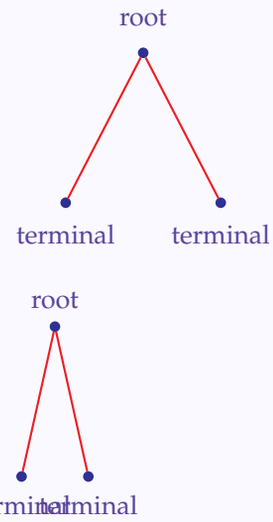
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

more detail in the section on bounding boxes.) The example below makes this clear:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{Violet}
  \pstree{\TBC*~[tnpos=a]{root}}
    {\TBC*~{terminal}\TBC*~{terminal}}

  \vspace{1cm}

  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \pstree{\TBC*\nput{u}{\pssucc}{root}}
    {\TBC*\nput{d}{\pssucc}{terminal}
     \TBC*\nput{d}{\pssucc}{terminal}}
\end{center}
```



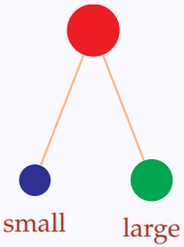
Again, the short-cut character ~ can be changed to any character *char* of our choice, using the command `\MakeShortTnput{`*char*`}`. We can even replace it by a *command* of our choice, say, `\tnput`, by defining `\MakeShortTnput{\tnput}`.

The distance between a node and its label can be adjusted using the `tnsep` parameter. Its default value is `labelsep`, which by default is 5 pt. The example below shows how this value can be locally reset to align labels:

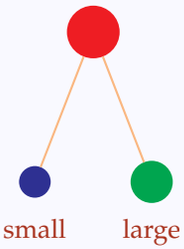
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \psset{linecolor=Apricot}
  \color{Mahogany}
  \pstree{\TC*[linecolor=Red,radius=10pt]}
    {\TC*[linecolor=Blue,radius=6pt]~{small}
     \TC*[linecolor=Green,radius=8pt]~{large}}

  \vspace{1cm}
```

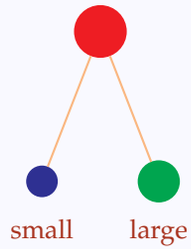



```
  \psset{linecolor=Apricot}
  \color{Mahogany}
  \pstree{\TC*[linecolor=Red,radius=10pt]}
    {\TC*[linecolor=Blue,radius=6pt]~[tnsep=7pt]{small}
     \TC*[linecolor=Green,radius=8pt]~{large}}
 \end{center}
```



An interesting feature of the `tnsep` parameter is that if it is set to a *negative* value, then the label is set at a distance equal to the absolute value of this, from the *center* of the node. Thus the same effect as in the bottom picture of the above example can be achieved by setting `tnsep=-13pt` globally.

```
\begin{center}
  \psset{linecolor=Apricot,tnsep=-13pt}
  \color{Mahogany}
  \pstree{\TC*[linecolor=Red,radius=10pt]}
    {\TC*[linecolor=Blue,radius=6pt]~{small}
     \TC*[linecolor=Green,radius=8pt]~{large}}
 \end{center}
```



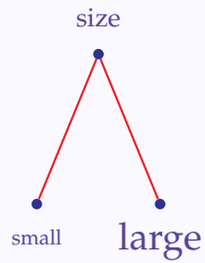
In the above example, the node-labels were not aligned by default, because of the different sizes of the nodes. It may also happen because of the different sizes of the labels. Look at this example:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

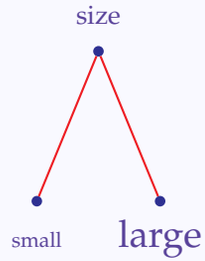
The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{Violet}
  \pstree{\TBC*~[tnpos=a]{size}}
    {\TBC*~{\footnotesize small}\TBC*~{\Large large}}
\end{center}
```



Here, the baselines of the labels are not aligned; and this is because the heights of the boxes containing the label texts are different. Now when a label is placed below a node in a pstree, it is given a minimum height set by the `tnheight` parameter (with default value, the TEX length `\ht\strutbox`). By changing its value suitably, we can align the labels as below:

```
\begin{center}
  \newlength{\largeht}
  \settoheight{\largeht}{\Large large}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{Violet}
  \pstree{\TBC*~[tnpos=a]{size}}
    {\TBC*~[tnheight=\largeht]{\footnotesize small}
     \TBC*~{\Large large}}
\end{center}
```



Similarly a label above a node is given a minimum depth `tndepth` (with default value `\dp\strutbox`) and this may have to be adjusted to align the labels as in the example below:

Online LaTeX Tutorial

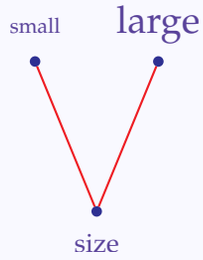
Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \newlength{\largedp}
  \settodepth{\largedp}{\Large large}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red,treemode=U}
  \color{Violet}
  \pstree{\TBC*~[tnpos=b]{size}}
    {\TBC*~[tndepth=\largedp]{\footnotesize small}
     \TBC*~{\Large large}}
\end{center}
```



As another example of setting `tnheight`, look at this:

```
\begin{center}
  \newlength{\fracht}
  \settoheight{\fracht}{$\dfrac{1}{2}$}
  \psset{treesep=0.22em,linecolor=Red}
   \color{Blue}
   \pstree{\Tp~[tnpos=a]{$1\frac{1}{2}$}}
     {\Tp~[tnheight=\fracht]{1}
      \Tp*[edge=none]~[tnheight=\fracht]{$+$}
      \Tp~{$\dfrac{1}{2}$}}
 \end{center}
```

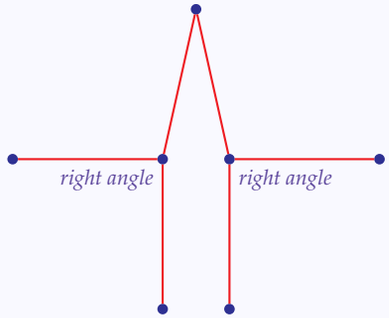
For labels above or below a node, the horizontal reference point of the node, that is the point directly above or below the (center of the) node is the center of (the box containing) the label. This can be changed using the `href` parameter. As explained in the discussion of `Rnode` in the last chapter, this is a *number*, the fraction of the horizontal distance of the reference point from the center of the box by half the length of the box, positive for the right half of the box and negative for the left half. See this example:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
`http://www.tug.org.in`

```
\begin{center}
  \footnotesize
  \itshape
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{Violet}
  \pstree{\TBC*}
    {\pstree{%
        \pstree[treemode=L]
          {\TBC*[href=1.2]~[tnsep=1pt]{right angle}}
          {\TBC*}}
       {\TBC*}
     \pstree{%
       \pstree[treemode=R]
         {\TBC*[href=-1.2]~[tnsep=1pt]{right angle}}
         {\TBC*}}
       {\TBC*}}
\end{center}
```

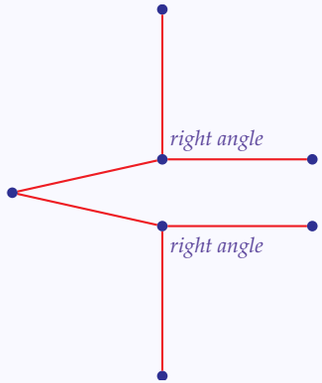


Similarly, for labels to the right or left of the node, the vertical reference point is, by default, the mid-point of the height of the label and this can be changed by using the `tnyref` parameter. Like `href`, this is also a number, but unlike `href`, it is the fraction of the vertical distance of the reference point from the *bottom* of the box by the *total height* of the box. The example below will make this clear:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to
tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \footnotesize
  \itshape
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psset{linecolor=Red}
  \color{Violet}
  \pstree[treemode=R]{\TBC*}
    {\pstree{%
      \pstree[treemode=U]
       {\TBC*[tnyref=-0.4]~[tnsep=1pt]{right angle}}
        {\TBC*}}
      {\TBC*}
     \pstree{%
       \pstree[treemode=D]
         {\TBC*[tnyref=1.5]~[tnsep=1pt]{right angle}}
         {\TBC*}}
       {\TBC*}}
\end{center}
```



Online LaTeX Tutorial

Part II – Graphics

PSTricks

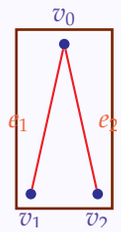
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

11.9. Bounding box

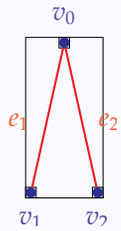
Every object in PostScript has a *bounding box*, which is the smallest rectangle containing the object. Now in computing the bounding box of trees, PSTricks does not take into account the labels. This becomes apparent when we try to frame trees as below:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psframebox[linecolor=Brown,linewidth=1pt]{%
    \color{Violet}
    \psset{linecolor=Red}
    \pstree{\TBC*\nput{u}{\pssucc}{$v_0$}}
      {\TBC*\nput{d}{\pssucc}{$v_1$}
       \tlput{\color{RedOrange} $e_1$}
       \TBC*\nput{d}{\pssucc}{$v_2$}
       \trput{\color{RedOrange} $e_2$}}}
\end{center}
```



The bounding box can be made visible by the `showbbox` parameter. Its default value is `false` and by setting it to `true`, we can see the bounding box, as in this example:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \color{Violet}
  \psset{linecolor=Red,showbbox=true}
  \pstree{\TBC*\nput{u}{\pssucc}{$v_0$}}
  {\TBC*\nput{d}{\pssucc}{$v_1$}
    \tlput{\color{RedOrange} $e_1$}
    \TBC*\nput{d}{\pssucc}{$v_2$}
    \trput{\color{RedOrange} $e_2$}}
\end{center}
```



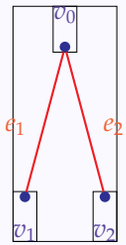
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

If we use the short-cut command for node-labels, then bounding box computation accommodates these labels, as shown below:

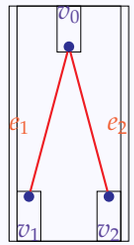
```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \color{Violet}
  \psset{linecolor=Red,showbbox=true}
  \pstree{\TBC*~[tnpos=a]{$v_0$}}
  {\TBC*~{$v_1$}
    \tlput{\color{RedOrange} $e_1$}
    \TBC*~{$v_2$}
    \trput{\color{RedOrange} $e_2$}}
\end{center}
```



Note the separate bounding box around each node label in this case.

To accommodate the edge-labels also, we must enlarge the bounding box. We can *specify* the dimensions of the bounding box, using the parameters, `bbl`, `bbr`, `bbh`, `bbd`, which correspond to the length to the left, length to the right, height and depth of the bounding box. all measured from the center of the box. There is also another set of four parameters `xbbl`, `xbbr`, `xbbh`, `xbbd`, which *increase* the default dimensions by the specified amount. See how we modify our example above:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \color{Violet}
  \psset{linecolor=Red,showbbox=true}
  \pstree[xbbl=3pt,xbbr=3pt]{\TBC*~[tnpos=a]{$v_0$}}
  {\TBC*~{$v_1$}
    \tlput{\color{RedOrange} $e_1$}
    \TBC*~{$v_2$}
    \trput{\color{RedOrange} $e_2$}}
\end{center}
```



Online LaTeX Tutorial

Part II – Graphics

PSTricks

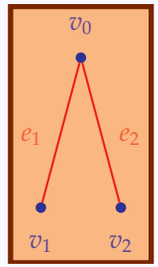
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

The picture shows both the original bounding box and the actual (extended) bounding box.

Now we can frame our tree as below:

```
\begin{center}
  \newpsobject{TBC}{TC}{linecolor=Blue,radius=2pt}
  \psframebox[linewidth=2pt,%
              linecolor=Brown,%
              fillstyle=solid,%
              fillcolor=Apricot]
              {\color{Violet}
               \psset{linecolor=Red}
               \pstree[xbbl=3pt,xbbr=3pt]
               {\TBC*~[tnpos=a]{$v_0$}}
               {\TBC*~{$v_1$}
                 \tlput{\color{RedOrange} $e_1$}
                 \TBC*~{$v_2$}
                 \trput{\color{RedOrange} $e_2$}}}
\end{center}
```



Online LaTeX Tutorial

Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in