

**4.1a** Give definitions of the following:

1. A path of length  $n$
2. The depth of a node
3. The height of a tree
4. The descendants of a node
5. The ancestors of a node

**4.1b** Fill in the blanks:

The collection of all descendants of a node forms a \_\_\_\_\_ .

The collection of all ancestors of a node forms a \_\_\_\_\_ .

**4.1c** A deque is implemented using a doubly linked list. Describe an  $\Theta(1)$  algorithm that will reverse the order of the elements in the deque. What would be required to achieve this runtime if a `Double_list` class had a similar implementation to our `Single_list` class?

**4.1d** Draw a tree that contains a node A where:

1. B is the parent of A,
2. C is a child of A,
3. D is a sibling of A,
4. E is an ancestor of A, and
5. F is a descendant but not a child of A.

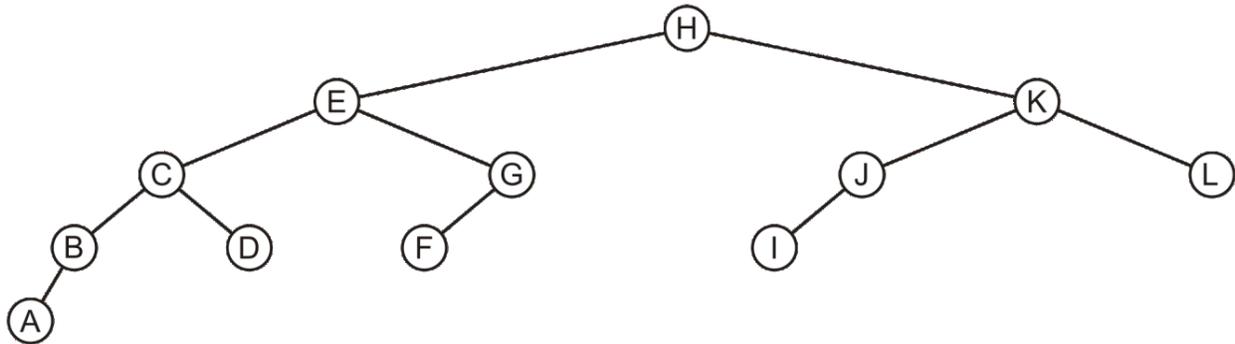
There are many answers to this question.

**4.1e** The root node is an ancestor of all nodes. Between any two nodes, there is therefore at least one common ancestor, although there may be more. The *lowest common ancestor* of a pair of nodes is the common ancestor which:

1. Has the greatest depth,
2. Also shares all the common ancestors of the two nodes, or
3. Has the shortest path to each of the two nodes.

Argue that these three definitions are equivalent.

**4.1f** Consider the following tree:



For any node within the tree, answer the following questions:

1. The height of the sub-tree rooted at that node, and
2. The depth of the node.

Find all paths of length three in this tree.

**4.1g** The following HTML is a cleaned up version of `zombo.com` (which holds the copyright). Draw this HTML document as a tree.

```
<html>
<head>
<title>ZOMBO</title>
</head>
<body bgcolor="#FFFFFF">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="http://active.macromedia.com/flash2/cabs/swflash.cab#version=4,0,0,0"
  id="inrozxa" width="100%" height="100%">
  <param name=movie value="welcomenew6.swf" />
  <param name=quality value="high" />
  <param name=bgcolor value="#FFFFFF" />
  <embed src="inrozxa.swf" quality="high" bgcolor="#FFFFFF" width="100%" height="100%"
    type="application/x-shockwave-flash"
    pluginspage=
      "http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash">
  </embed>
</object>
</body>
</html>
```

A tag of the form `<tag />` is, in a sense, equivalent to `<tag></tag>`. I say “in a sense” because the first indicates no content, while the second indicates a content equivalent to an empty string `""`.

Some of the tags may contain *attributes*, for example, the `<body>` tag has the attribute `bgcolor` and the attribute has the value `"#FFFFFF"` (white). You do not have to indicate attributes in your tree.

**4.1h** The following MATHML describes an equation; what is it?

```
<math xmlns='http://www.w3.org/1998/Math/MathML'>
<semantics>
<mrow><mrow>
  <munderover><mo>&Integral;</mo><mn>0</mn><mi>x</mi></munderover>
  <mrow>
    <mi>sin</mi><mo>&ApplyFunction;</mo><mfenced><mi>&xi;</mi></mfenced>
  </mrow><mo>&InvisibleTimes;</mo>
  <mrow><mo>&DifferentialD;</mo><mi>&xi;</mi></mrow>
</mrow><mo>=</mo>
<mrow><mn>1</mn><mo>-</mo><mrow>
  <mi>cos</mi><mo>&ApplyFunction;</mo><mfenced><mi>x</mi></mfenced>
</mrow></mrow></mrow>
<annotation-xml encoding='MathML-Content'>
  <apply><eq/>
    <apply><int/>
      <bvar><ci>xi</ci></bvar>
      <lowlimit><cn>0</cn></lowlimit>
      <uplimit><ci>x</ci></uplimit>
      <apply><sin/><ci>xi</ci></apply>
    </apply>
    <apply><minus/>
      <cn>1</cn>
      <apply><cos/><ci>x</ci></apply>
    </apply>
  </apply>
</annotation-xml>
<annotation encoding='Maple'>Int(sin(xi),xi = 0 .. x) = 1-cos(x)</annotation>
</semantics></math>
```