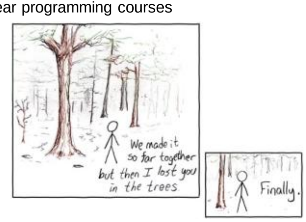
Trees

Outline

In this topic, we will cover:
- Definition of a tree data structure and its components
- Concepts of:
  - Root, internal, and leaf nodes
  - Parents, children, and siblings
  - Paths, path length, height, and depth
  - Ancestors and descendants
  - Ordered and unordered trees
  - Subtrees
- Examples
  - XHTML and CSS

The Tree Data Structure

Trees are the first data structure different from what you've seen in your first-year programming courses

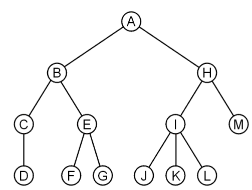


http://xkcd.com/71/

4.1.1

Trees

A rooted tree data structure stores information in *nodes*
- Similar to linked lists:
  - There is a first node, or *root*
  - Each node has variable number of references to successors
  - Each node, other than the root, has exactly one node pointing to it

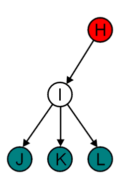


4.1.1.1

Terminology

All nodes will have zero or more child nodes or *children*
- I has three children: J, K and L

For all nodes other than the root node, there is one parent node
- H is the parent I

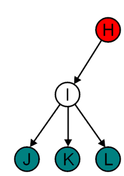


4.1.1.1

Terminology

The *degree* of a node is defined as the number of its children:
deg(I) = 3

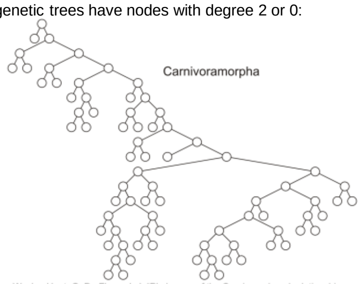
Nodes with the same parent are *siblings*
- J, K, and L are siblings



The tree data structure 7

4.1.1.1 **Terminology**

Phylogenetic trees have nodes with degree 2 or 0:



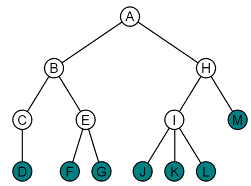
Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure 8

4.1.1.1 **Terminology**

Nodes with degree zero are also called *leaf nodes*

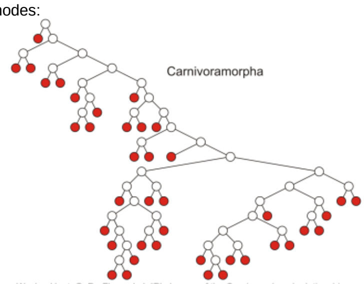
All other nodes are said to be *internal nodes*, that is, they are internal to the tree



The tree data structure 9

4.1.1.1 **Terminology**

Leaf nodes:

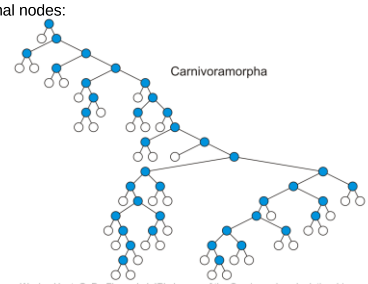


Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure 10

4.1.1.1 **Terminology**

Internal nodes:



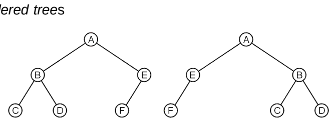
Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure 11

4.1.1.2 **Terminology**

These trees are equal if the order of the children is ignored

– *unordered trees*



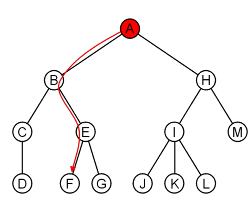
They are different if order is relevant (*ordered trees*)

– We will usually examine ordered trees (linear orders)
– In a hierarchical ordering, order is not relevant

The tree data structure 12

4.1.1.3 **Terminology**

The shape of a rooted tree gives a natural flow from the *root node*, or just *root*



The tree data structure 13

4.1.1.3

Terminology

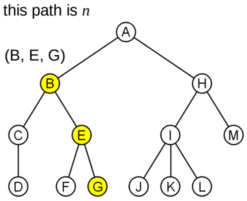
A path is a sequence of nodes

$(a_0, a_1, \ldots, a_n)$

where a_{k+1} is a child of a_k is

The length of this path is n

E.g., the path (B, E, G) has length 2

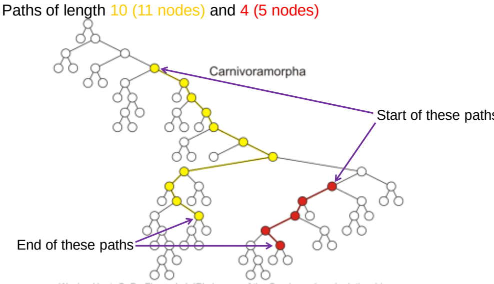


The tree data structure 14

4.1.1.3

Terminology

Paths of length 10 (11 nodes) and 4 (5 nodes)



Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure 15

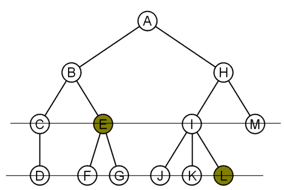
4.1.1.3

Terminology

For each node in a tree, there exists a unique path from the root node to that node

The length of this path is the *depth* of the node, *e.g.*,

- E has depth 2
- L has depth 3

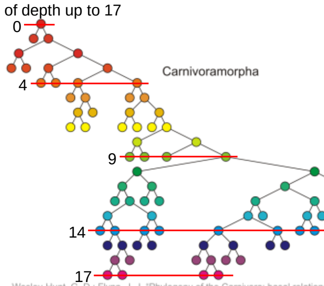


The tree data structure 16

4.1.1.3

Terminology

Nodes of depth up to 17



Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure 17

4.1.1.3

Terminology

The *height* of a tree is defined as the maximum depth of any node within the tree

The height of a tree with one node is 0

- Just the root node

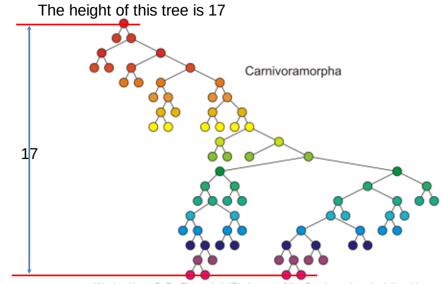
For convenience, we define the height of the empty tree to be -1

The tree data structure 18

4.1.1.3

Terminology

The height of this tree is 17



Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure

19

4.1.1.4

Terminology

If a path exists from node a to node b:
- a is an *ancestor* of b
- b is a *descendent* of a

Thus, a node is both an ancestor and a descendant of itself
- We can add the adjective *strict* to exclude equality: a is a *strict descendent* of b if a is a descendant of b but $a \neq b$

The root node is an ancestor of all nodes

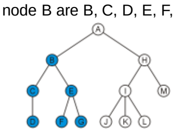
The tree data structure

20

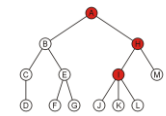
4.1.1.4

Terminology

The descendants of node B are B, C, D, E, F, and G:



The ancestors of node I are I, H, and A:



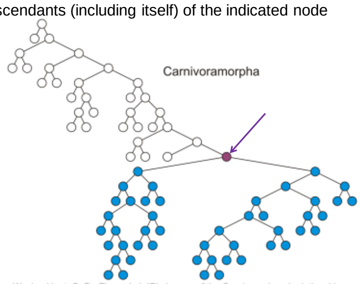
The tree data structure

21

4.1.1.4

Terminology

All descendants (including itself) of the indicated node



Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

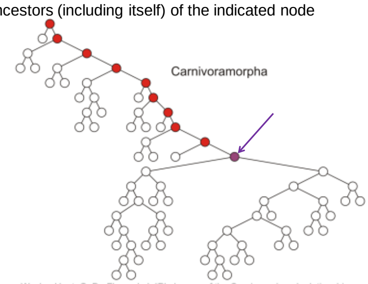
The tree data structure

22

4.1.1.4

Terminology

All ancestors (including itself) of the indicated node



Wesley-Hunt, G. D.; Flynn, J. J. "Phylogeny of the Carnivora: basal relationships among the Carnivoramorphans, and assessment of the position of 'Miacoidea'

The tree data structure

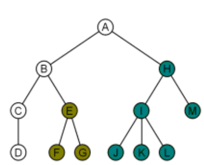
23

4.1.2

Terminology

Another approach to a tree is to define the tree recursively:
- A degree-0 node is a tree
- A node with degree n is a tree if it has n children and all of its children are disjoint trees (*i.e.*, with no intersecting nodes)

Given any node a within a tree with root r, the collection of a and all of its descendants is said to be a *subtree of the tree with root* a



The tree data structure

24

4.1.3

Example: XHTML and CSS

The XML of XHTML has a tree structure

Cascading Style Sheets (CSS) use the tree structure to modify the display of HTML

The tree data structure 25

4.1.3

Example: XHTML and CSS

Consider the following XHTML document

```
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>This is a <u>Heading</u></h1>

        <p>This is a paragraph with some
        <u>underlined</u> text.</p>
    </body>
</html>
```

The tree data structure 26

4.1.3

Example: XHTML and CSS

Consider the following XHTML document

```
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>This is a <u>Heading</u></h1>

        <p>This is a paragraph with some
        <u>underlined</u> text.</p>
    </body>
</html>
```

title

heading

body of page

underlining

paragraph

The tree data structure 27

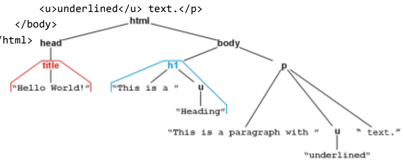
4.1.3

Example: XHTML and CSS

The nested tags define a tree rooted at the HTML tag

```
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>This is a <u>Heading</u></h1>

        <p>This is a paragraph with some
        <u>underlined</u> text.</p>
    </body>
</html>
```

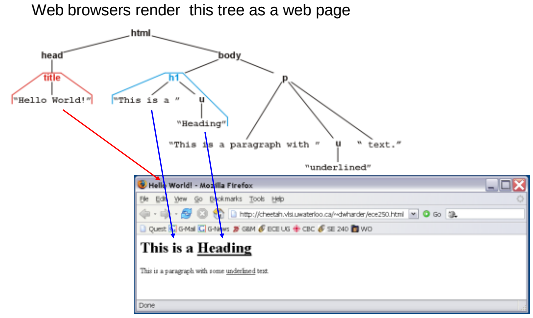


The tree data structure 28

4.1.3

Example: XHTML and CSS

Web browsers render this tree as a web page



The tree data structure 29

4.1.3

Example: XHTML and CSS

XML tags `<tag>...</tag>` must be nested

For example, to get the following effect:

<u>1 2 3 **4 5 6**</u>**7 8 9**

you may use

```
<u>1 2 3 <b>4 5 6</b></u><b> 7 8 9</b>
```

You may not use:

```
<u>1 2 3 <b>4 5 6</u> 7 8 9</b>
```

The tree data structure 30

4.1.3.1

Example: XHTML and CSS

Cascading Style Sheets (CSS) make use of this tree structure to describe how HTML should be displayed

- For example:

```
<style type="text/css">
    h1 { color:blue; }
</style>
```

indicates all text/decorations <u>descendant</u> from an **h1** header should be blue

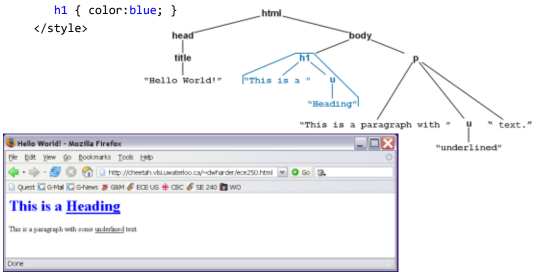
The tree data structure 31

4.1.3.1

Example: XHTML and CSS

For example, this style renders as follows:

```
<style type="text/css">
   h1 { color:blue; }
</style>
```



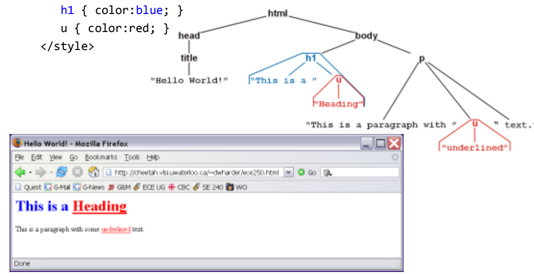
The tree data structure 32

4.1.3.1

Example: XHTML and CSS

For example, this style renders as follows:

```
<style type="text/css">
   h1 { color:blue; }
   u { color:red; }
</style>
```



The tree data structure 33

4.1.3.1

Example: XHTML and CSS

Suppose you don't want underlined items in headers (h1) to be red

- More specifically, suppose you want any underlined text within paragraphs to be red

That is, you only want text marked as `<u>text</u>` to be underlined if it is a descendant of a `<p>` tag

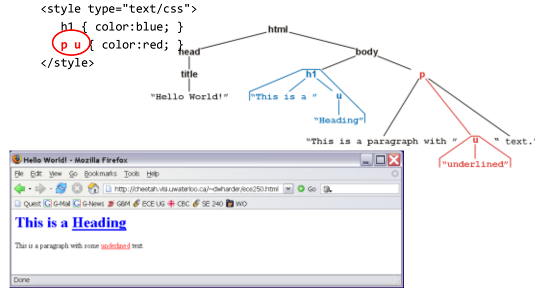
The tree data structure 34

4.1.3.1

Example: XHTML and CSS

For example, this style renders as follows:

```
<style type="text/css">
   h1 { color:blue; }
  p u { color:red; }
</style>
```



The tree data structure 35

4.1.3.1

Example: XHTML and CSS

You can read the second style

```
<style type="text/css">
   h1 { color:blue; }
   p u { color:red; }
</style>
```

as saying "text/decorations descendant from the underlining tag (<u>) which itself is a descendant of a paragraph tag should be coloured red"

The tree data structure 36

4.1.3.1

Example: XML

In general, any XML can be represented as a tree

- All XML tools make use of this feature
- Parsers convert XML into an internal tree structure
- XML transformation languages manipulate the tree structure
  - *E.g.*, XMLT

The tree data structure

37

4.1.3.1

MathML: $x^2 + y^2 = z^2$

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <semantics>
    <mrow><mrow><msup><mi>x</mi><mn>2</mn></msup><mo>+</mo>
    <msup><mi>y</mi><mn>2</mn></msup></mrow>
    <mo>=</mo><msup><mi>z</mi><mn>2</mn></msup></mrow>
    <annotation-xml encoding="MathML-Content">
      <apply><eq/>
        <apply><plus/>
          <apply><power/><ci>x</ci><cn>2</cn></apply>
          <apply><power/><ci>y</ci><cn>2</cn></apply>
        </apply>
        <apply><power/><ci>z</ci><cn>2</cn></apply>
      </apply>
    </annotation-xml>
    <annotation encoding="Maple">x^2+y^2 = z^2</annotation>
  </semantics>
</math>
```

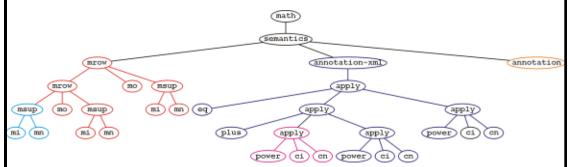
The tree data structure

38

4.1.3.1

MathML: $x^2 + y^2 = z^2$

The tree structure for the same MathML expression is



The tree data structure

39

4.1.3.1

MathML: $x^2 + y^2 = z^2$

Why use 500 characters to describe the equation

$$x^2 + y^2 = z^2$$

which, after all, is only twelve characters (counting spaces)?

The root contains three children, each different codings of:
- How it should look (presentation),
- What it means mathematically (content), and
- A translation to a specific language (Maple)

The tree data structure

40

Summary

In this topic, we have:
- Introduced the terminology used for the tree data structure
- Discussed various terms which may be used to describe the properties of a tree, including:
  - root node, leaf node
  - parent node, children, and siblings
  - ordered trees
  - paths, depth, and height
  - ancestors, descendants, and subtrees
- We looked at XHTML and CSS

The tree data structure

41

References

[1] Donald E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, 3rd Ed., Addison Wesley, 1997, §2.2.1, p.238.