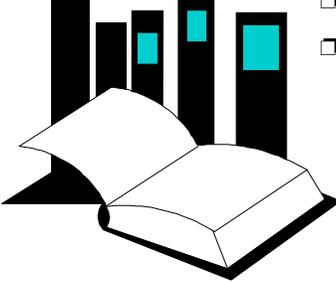


Complete Binary Trees



- Chapter 10 introduces **trees**.
- This presentation illustrates the simplest kind of trees: **Complete Binary Trees**.

**Data Structures
and Other Objects
Using C++**

1

This lecture is an introduction to trees, illustrating basic terminology for binary trees, and focusing on complete binary

Before this lecture, students should have a good understanding of the simpler linear data structures of lists, stacks, and queues.

Binary Trees

- ❑ A binary tree has **nodes**, similar to nodes in a linked list structure.
- ❑ **Data** of one sort or another may be stored at each node.
- ❑ But it is the **connections** between the nodes which characterize a binary tree.

In many ways, a tree is like the other structures you have seen: A tree consists of nodes, and each node can contain data of one sort or another.

Binary Trees

- ❑ A binary tree has **nodes**, similar to nodes in a linked list structure.
- ❑ **Data** of one sort or another may be stored at each node.
- ❑ But it is the **connections** between the nodes which characterize a binary tree.

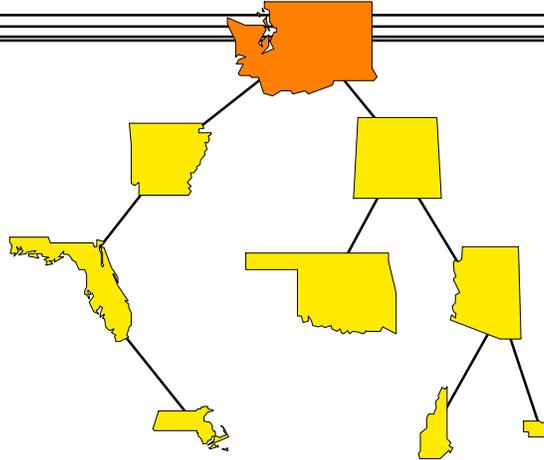


An example can illustrate how the connections work

But, unlike a linked list, the connections between the nodes are more than a simple one-to-another progression. An example can illustrate the connections in a binary tree.

A Binary Tree of States

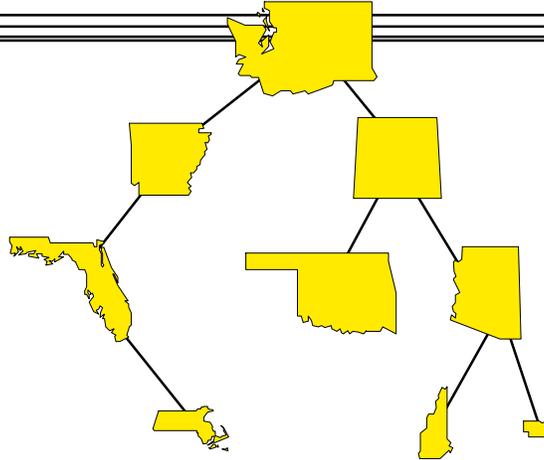
Each tree has a special node called its **root**, usually drawn at the top.



The first special kind of node is the root. Computer scientists always draw this special node at the top. (I think they must have flunked botany.)

A Binary Tree of States

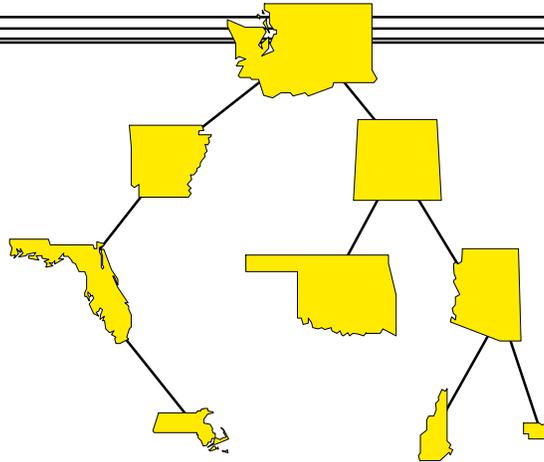
Each node is permitted to have two links to other nodes, called the **left child** and the **right child**.



Each node in a binary tree is permitted to have two links downward to other nodes, called the left child and the right child.

A Binary Tree of States

Each node is permitted to have two links to other nodes, called the **left child** and the **right child**.

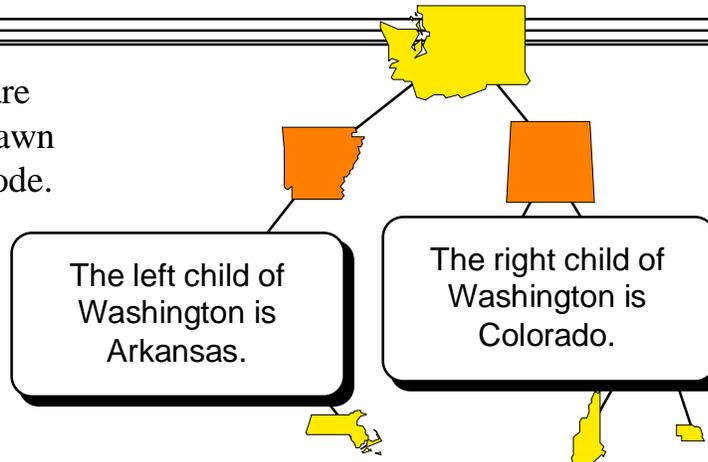


The fact that we allow two links to children is the reason that these trees are called binary trees. "Binary" means "two parts".

Later you will see more general trees where the number of children is not restricted to two.

A Binary Tree of States

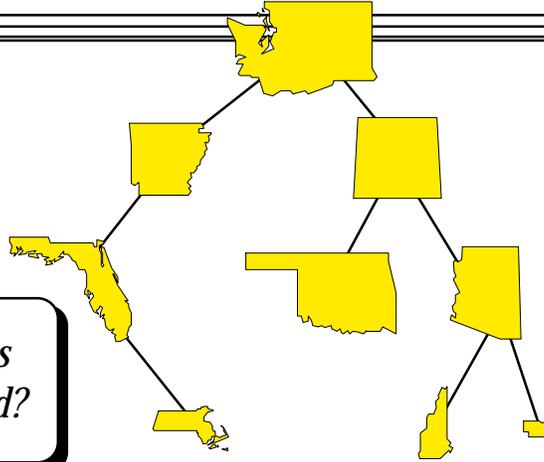
Children are usually drawn below a node.



But for binary trees, each node has at most two children. For example, the left child of Washington is Arkansas. The right child of Washington is Colorado.

A Quiz

Some nodes
have only one
child.

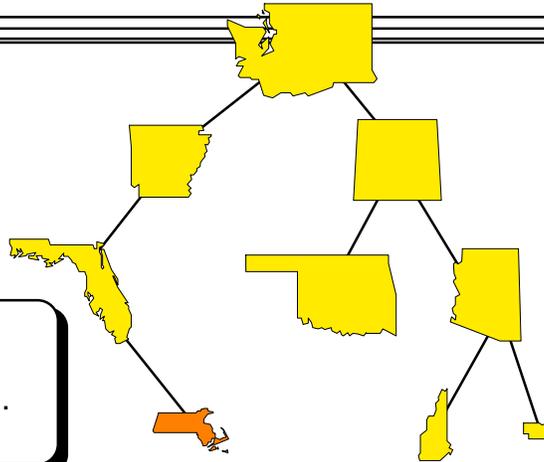


*Which node has
only a right child?*

Here is a quiz: Find a node that has a right child, but no left child...

A Quiz

Some nodes
have only one
child.



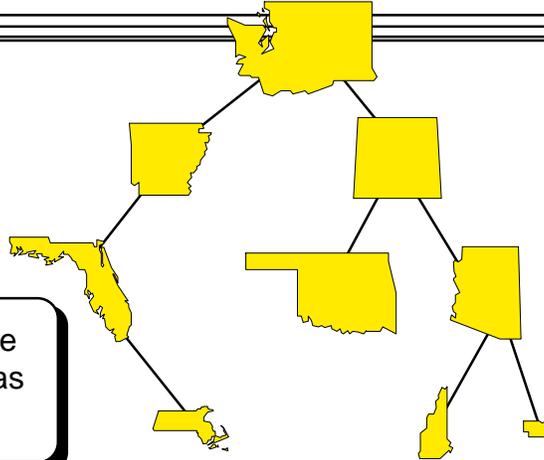
Florida has
only a right child.

In this example, Massachusetts is the right child of Florida, and Florida has no left child.

A Binary Tree of States

Each node is called the **parent** of its children.

Washington is the parent of Arkansas and Colorado.

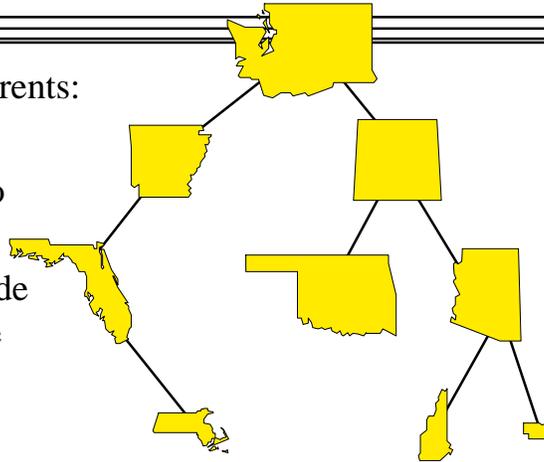


We can also refer to the parent of a node, which is the upward link. For example, Washington is the parent of both Arkansas and Colorado.

A Binary Tree of States

Two rules about parents:

- ❶ The root has no parent.
- ❷ Every other node has exactly one parent.



There are two rules about parents in any tree:

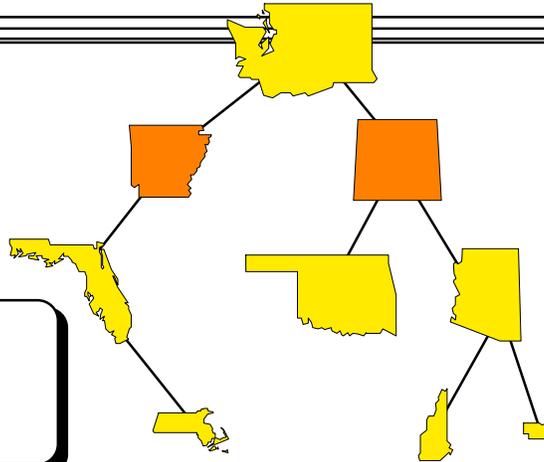
1. The root never has a parent.
2. Every other node has exactly one parent.

There is also a related rule which is not written here, but is part of the definition of a tree: If you start at the root, there is always one way to get from the root to any particular node by following a sequence of downward links (from a parent to its child).

A Binary Tree of States

Two nodes with the same parent are called siblings.

Arkansas and Colorado are siblings.



Two nodes that have the same parent are called siblings, as shown here.

In a binary tree, a node has at most one sibling.

Complete Binary Trees

A **complete** binary tree is a special kind of binary tree which will be useful to us.

The rest of this lecture demonstrates a special kind of binary tree called a complete binary tree. The tree has several applications, and is also special because it is extremely easy to implement.

Complete Binary Trees

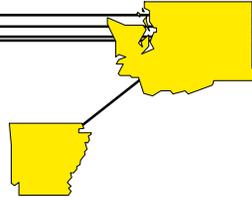
A complete binary tree is a special kind of binary tree which will be useful to us.

When a complete binary tree is built, its first node must be the root.

When a complete binary tree is built, its nodes are generally added one at a time. As with any tree, the first node must be the root.

Complete Binary Trees

The second node of a complete binary tree is always the left child of the root...

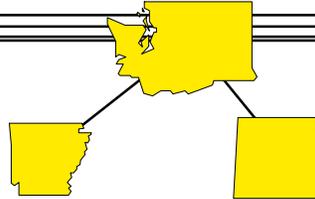


With a complete binary tree, the second node must be the left child of the root.

Complete Binary Trees

The second node of a complete binary tree is always the left child of the root...

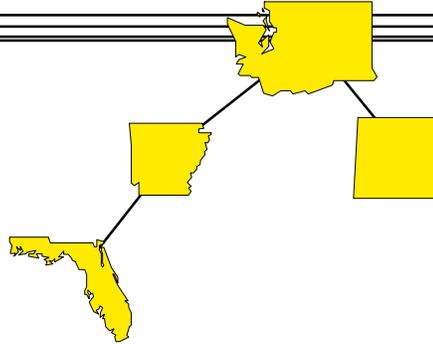
... and the third node is always the right child of the root.



The next node must be the right child of the root.

Complete Binary Trees

The next nodes must always fill the next level from **left to right**.

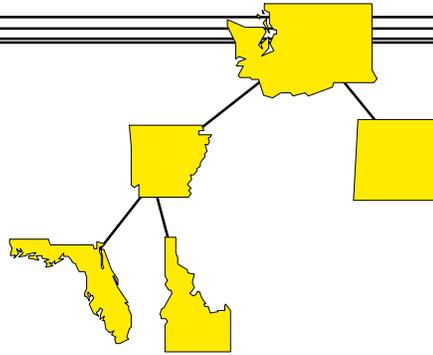


And so we continue, adding nodes. But in a complete binary tree, the way that we add nodes is restricted: Nodes must completely fill each level from left-to-right before proceeding to the next level.

Following this rule, where would the fifth node have to be placed?

Complete Binary Trees

The next nodes must always fill the next level from **left to right**.

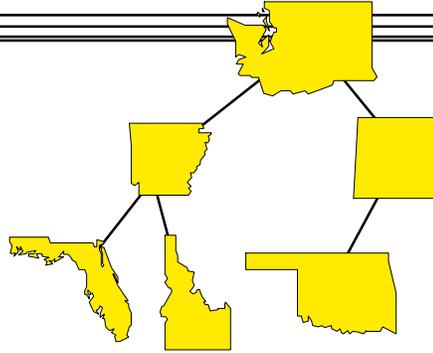


The fifth node must go in the next available spot in this level, which is the right child of Arkansas.

By the way, there is also an interesting story about Idaho. Apparently congress decreed that the border between Idaho and Montana was to be the continental divide. So the surveyors started at Yellowstone Park, mapping out the border, but at “Lost Trail Pass” they made a mistake and took a false divide. They continued along the false divide for several hundred miles until they crossed over a ridge and saw the Clark Fork River cutting right across what they thought was the divide. At this point, they just gave up and took the rest of the border straight north to Canada. So those of you who come from Kalispell and Missoula Montana should really be Idahoians.

Complete Binary Trees

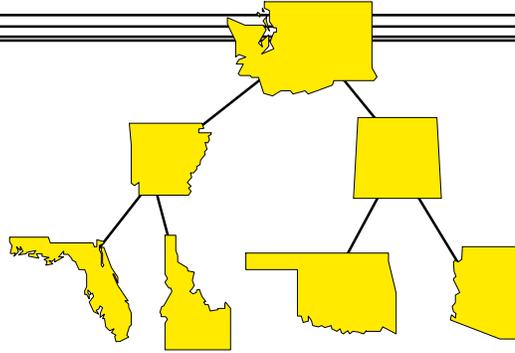
The next nodes must always fill the next level from **left to right**.



Then we can add the left child of Colorado...

Complete Binary Trees

The next nodes must always fill the next level from **left to right**.

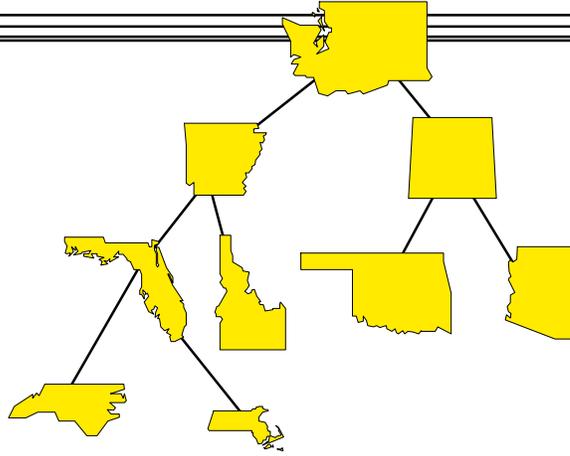


...and the right child of Colorado.

Where would the next node after this go?...

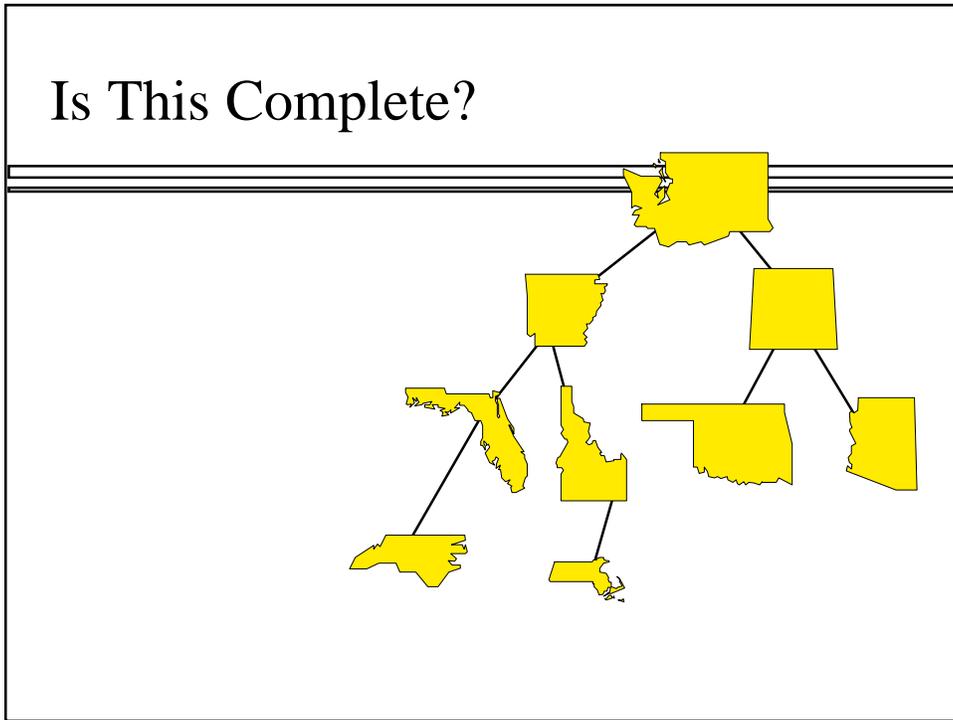
Complete Binary Trees

The next nodes must always fill the next level from **left to right**.



Then the right child of Florida.

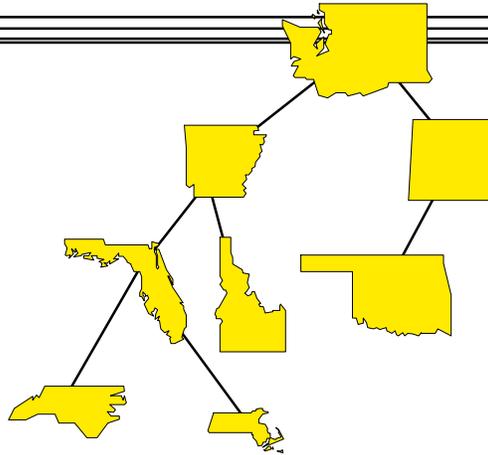
Is This Complete?



Just to check your understanding, is this binary tree a complete binary tree?

No, it isn't since the nodes of the bottom level have not been added from left to right.

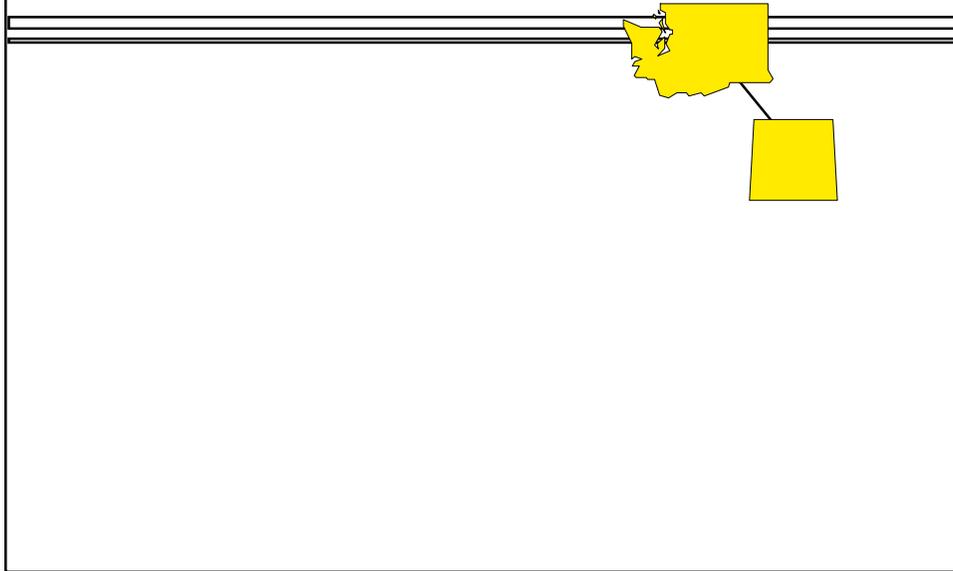
Is This Complete?



Is this complete?

No, since the third level was not completed before starting the fourth level.

Is This Complete?



This is also not complete since Washington has a right child but no left child.

Is This Complete?

30



But this binary tree is complete. It has only one node, the root.

Is This Complete?

Yes!

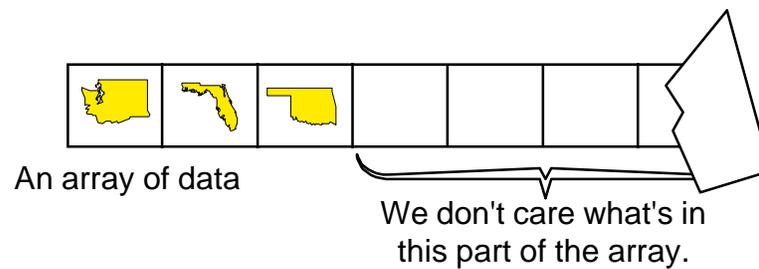
- ✓ It is called the empty tree, and it has no nodes, not even a root.

This binary tree is also complete, although you might have to squint to see it. You see, this binary tree has no nodes at all. It is called the empty tree, and it is considered to be a complete binary tree.

Implementing a Complete Binary Tree

- We will store the data from the nodes in a partially-filled array.

3 An integer to keep track of how many nodes are in the tree

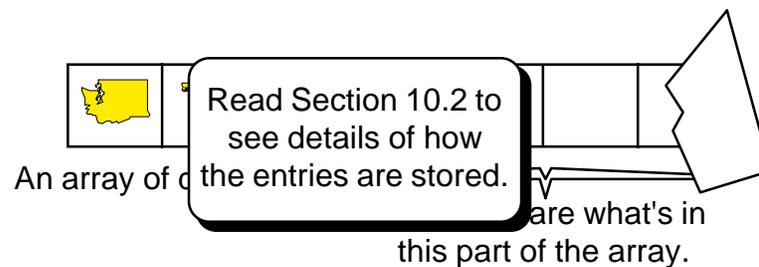


There is one last feature of complete binary trees that you need to learn: How to implement a complete binary tree using a partially-filled array. With this implementation, the data from each node is placed in one component of a partially-filled array. The primary thing we need to keep track of is where in the array does each node belong?

Implementing a Complete Binary Tree

- We will store the data from the nodes in a partially-filled array.

3 An integer to keep track of how many nodes are in the tree



The solution to this problem is in Section 10.2 of the text, which discusses several different ways to implement binary trees. The simplest implementation uses a partially-filled array, as suggested by this picture, but you should be warned: The array implementation is only practical for complete binary trees.



Summary

- ❑ Binary trees contain nodes.
- ❑ Each node may have a left child and a right child.
- ❑ If you start from any node and move upward, you will eventually reach the root.
- ❑ Every node except the root has one parent. The root has no parent.
- ❑ Complete binary trees require the nodes to fill in each level from left-to-right before starting the next level.

A quick summary . . .

Presentation copyright 1997 Addison Wesley Longman,
For use with *Data Structures and Other Objects Using C++*
by Michael Main and Walter Savitch.

Some artwork in the presentation is used with permission from Presentation Task Force
(copyright New Vision Technologies Inc) and Corel Gallery Clipart Catalog (copyright
Corel Corporation, 3G Graphics Inc, Archive Arts, Cartesia Software, Image Club
Graphics Inc, One Mile Up Inc, TechPool Studios, Totem Graphics Inc).

Students and instructors who use *Data Structures and Other Objects Using C++* are welcome
to use this presentation however they see fit, so long as this copyright notice remains
intact.



Feel free to send your ideas to:

Michael Main

main@colorado.edu