Andrew Plum

Cs 121

5/2/22

Week 16 Notes

* Print order
  + Infix
    - Tree::print(node \*current){

If(current == NULL)

return;

}

Print(left);

Cout << current->data << endl;

Print(right);

}

* + - Tree::print(node \*current){

If(current->left != NULL)

print(left);

}

Cout << current->data << endl;

If(current->right != NULL)

print(right);

}

}

* + - Node::print(){

If(left != NULL){

Left->print();

}

Cout << data << “, “ << endl;

If(right != NULL){

right->print();

}

}

* ReTRIEval Tree (Trie for short – pronounced like try)
  + Application
    - Is really good and faster and simpler than binary search trees than when doing usernames data base
    - Not used to print names, just searches
  + abcdef



* + Node in a trie has
    - N branches
    - Bool to mark end of entries
  + Have to have an end = true to terminate the string
    - So if you have Jackson in the trie this doesn’t mean you have Jack
      * You could possibly have Jack in the trie but the string has to be terminated by end = true
  + Const int length = 26;
  + First Method:
    - Class node{

Public:

Node \*children; // USE A LINK LIST OF NODE POINTERS

Bool endofentry;

Node();

};

Node::node(){

For(int i … length){

Children[i] = NULL;

}

}

* + - Implement link list using template I already have or from standard template library
  + Second Method:
    - Class node{

Public:

Node \*children[length];

Bool endofentry;

Node();

};

Node::node(){

For(int i … length){

Children[i] = NULL;

}

}