Plan for Today

Complexity of algorithm from Wednesday

Tree traversals
- depth-first traversal
- pre-order, post-order, and in-order

Visitor Design Pattern
- why?
- what?
- traversing trees with a visitor design pattern

SableCC’s Visitor Design Pattern
- terminology
- using it

Example for Tree Traversals

Visitor Design Pattern

Situation
- Want to perform some processing on all items in a data structure
- Will be adding many different ways to process items, different features
- Will not be changing the classes of the data structure itself much

Possibilities
- For each functionality add a method to all of the classes
  - Example of this in PA5, outputDot
  - Each new functionality is spread over multiple files
  - Sometimes can’t do it
- Use a large if-then-else statement in visit method
  - pro: keeps all the code for the feature in one place
  - con: can be costly and involve lots of casting
- Visitor design pattern

Borrowed SableCC Visitor Design Pattern

```java
BuildSymTable buildSTvisitor = new BuildSymTable(linesToNodes); 
ast.apply(buildSTvisitor); 
SymTable globalST = buildSTvisitor.getSymTable();

// in class VarDecl
public void apply(Switch sw) 
{ ((Analysis) sw).caseVarDecl(this); }

// in class DepthFirstAdapter
public void inVarDecl(VarDecl node) { defaultIn(node); }
public void outVarDecl(VarDecl node) { defaultOut(node); }
public void caseAndExp(VarDecl node) {
  inVarDecl(node);
  if(node.getType() != null) { node.getType().apply(this); }
  if(node.getName() != null) { node.getName().apply(this); }
  outVarDecl(node);
}
```
FAQ

How do I associate data with a node in the AST if I can’t add fields to the node classes?

What if I want to do the same thing on each node?

What if I only need to do something on certain nodes?

Shouldn’t the visit/case methods have return values?