

Recursion

Robotic Insect Problem

Robotic Insect Problem

- Writing a program to control an insect robot
- Robot is climbing a tree from base of tree upwards
- Looking for food
- Going to eat all the food on the tree

Given these functions to use:

int IsFoodHere(); /* returns a 1 when there is food where robot is located; returns 0 otherwise */

void EatFood(); /* eat food if there is food; see below */

void TakeLeftBranch(); /* takes left branch if it exists*/

void TakeRightBranch(); /* takes right branch if it exists */

void ReturnFromBranch(); /* returns back to prior branch choice point */

int IsBranch(); /* returns 1 if there is a left and a right branch; returns 0 otherwise */

Robotic Insect (Con't)

- Climb only simple trees
 - Both left and right branch or end of branch
- **Goal:** Write a recursive function, using above functions, so insect eats all food on tree
- Note that if you call *EatFood()* and *IsFoodHere()* was not true at that point, the insect will eat bark, and die.

Robotic Insect (Con't)

- Start at bottom of tree. After moving with *TakeLeftBranch()* or *TakeRightBranch()*, insect is at another branching point (if there were branches) or at end of branch

```
void robotInsect() {  
    if (FoodIsHere()) EatFood();  
    if (IsBranch()) {  
        TakeLeftBranch();  
        RobotInsect();  
        ReturnFromBranch();  
        TakeRightBranch();  
        RobotInsect();  
        ReturnFromBranch();  
    }  
}
```

Robot Insect: Find First Piece of Food

- **Goal:** Write a recursive function, using above functions, so insect eats just one piece of food on tree
 - After insect done eating single piece of food, returns to base of tree

New Algorithm

```
int RobotInsect2() {
    int found = 0;
    if (FoodIsHere()) {
        EatFood(); found = 1;
    }
    if ((! found) && (IsBranch())) {
        TakeLeftBranch(); found = RobotInsect2();
        ReturnFromBranch();
        if (! found) {
            TakeRightBranch(); found = RobotInsect();
            ReturnFromBranch();
        }
    }
    return found;
}
```