LogicVis

By Candice Miao, Leo Gao, Jed Chen, Glenn Zhang, Andrew Liu

Introduction

People have trouble understanding code in many situations

- Learning to code
- Reading others' code

Recursion is one of the topics that are difficult to understand

Use visualization to help understanding code, especially recursion

Problem: Understanding Recursion

Study shows that only 13% of the students are able to completely understand recursion.

Reasons

1. Students are not using abstractions well enough
2. Lack of a proper methodology to represent a recursive solution

Scholtz, Tamarisk Lurlyn, and Ian Sanders. “Mental Models of Recursion.” *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE '10*, 2010, doi:10.1145/1822090.1822120.

Motivation

- **Facilitate the learning experience of recursion for new recursion learners in Computer Science**
- Make understanding foreign code easier
- Help programmer debug

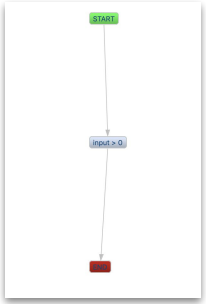
Previous Approaches

Code Visualization Tools
- Eclipse CFG Generator
  - No indication of recursion or even function call

Recursion Trackers
- VisuAlgo
  - No detail about the function is presented

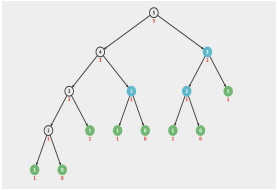
ChiQat-Tutor system
- A system that helps visualize the recursive calls
- Only visualizes pre-defined cases

```
public static int test(int input) {
    if (input > 0) {
        return 2 * test(input - 2);
    } else {
        return input - 10;
    }
}
```

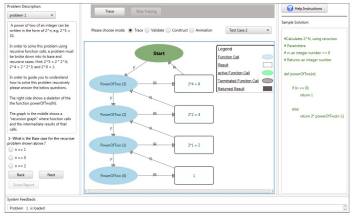


(Eclipse CFG Generator)

```
if (n <= 1) /* base case */
  return n;
else /* recursive caseS */
  return f(n-1) + f(n-2);
```



(VisuAlgo)



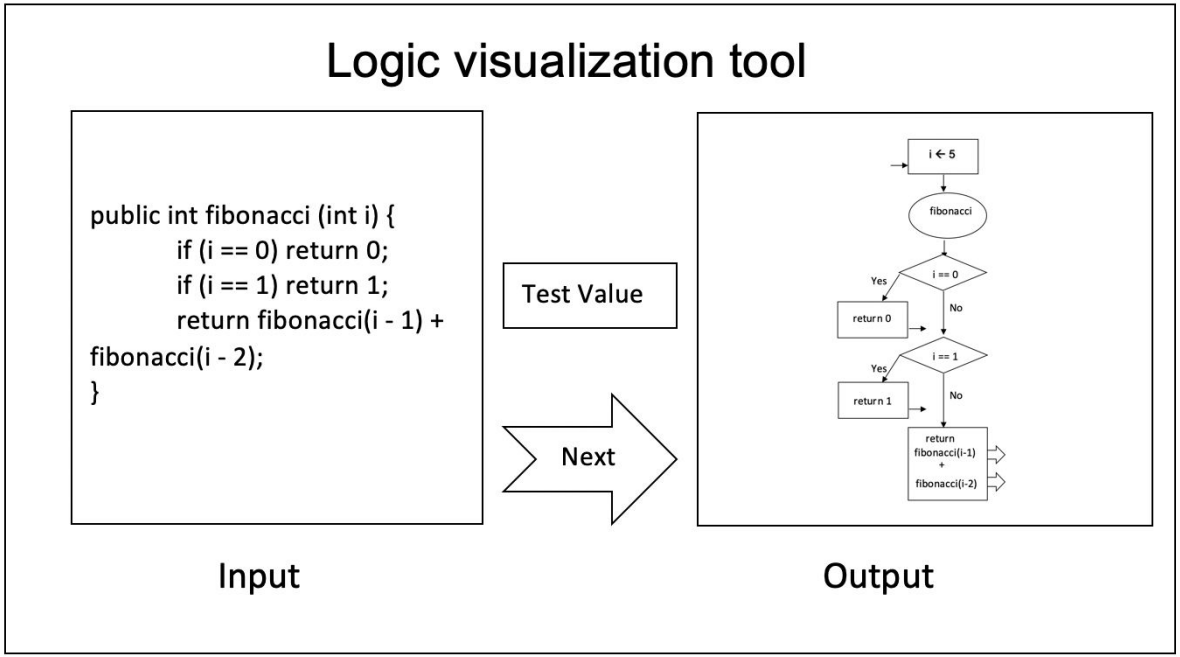
(ChiQat-Tutor
)

AlZoubi, Omar & Fossati, Davide & Di Eugenio, Barbara & Green, Nick & Alizadeh, Mehrdad & Harsley, Rachel. (2015). A Hybrid Model for Teaching Recursion. 10.1145/2808006.2808030.

Outline

- **Our Solution**
  - Research
  - Challenges and Risks

Approach

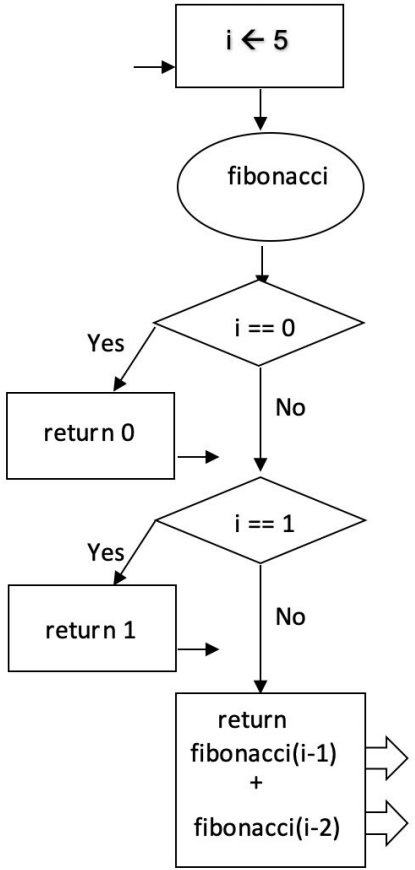

Logic visualization tool
public int fibonacci (int i) {
if (i == 0) return 0;
if (i == 1) return 1;
return fibonacci(i - 1) +
fibonacci(i - 2);
}
Test Value
Next
i ← 5
fibonacci
i == 0
Yes
No
return 0
i == 1
Yes
No
return 1
return fibonacci(i-1) + fibonacci(i-2)
Input
Output

Why it is useful

- Visualization is easier to understand
- Our solution not only indicates the function call but also visualizes the specific call iteration
- Our solution provides the parameters of each function call

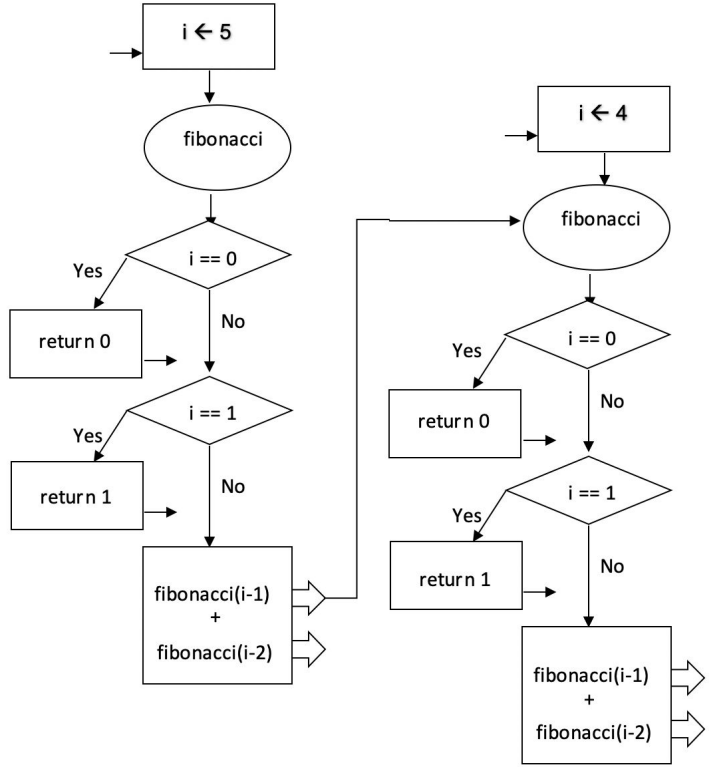
Example

```
public int fibonacci (int i) {
        if (i == 0) return 0;
        if (i == 1) return 1;
        return fibonacci(i - 1) + fibonacci(i - 2);
}
```



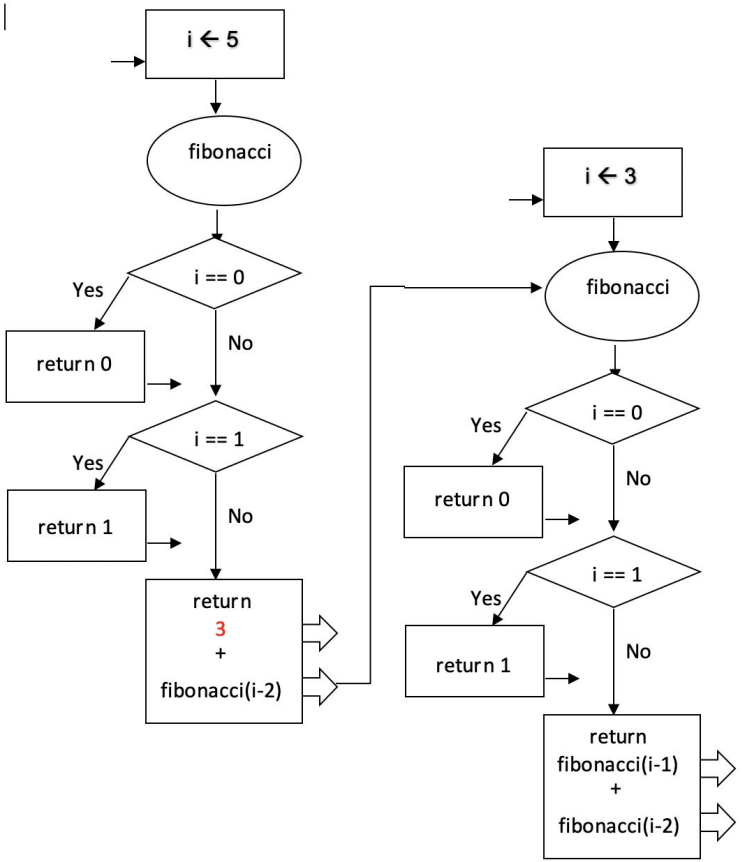
Example

```
public int fibonacci (int i) {
        if (i == 0) return 0;
        if (i == 1) return 1;
        return fibonacci(i - 1) + fibonacci(i - 2);
}
```

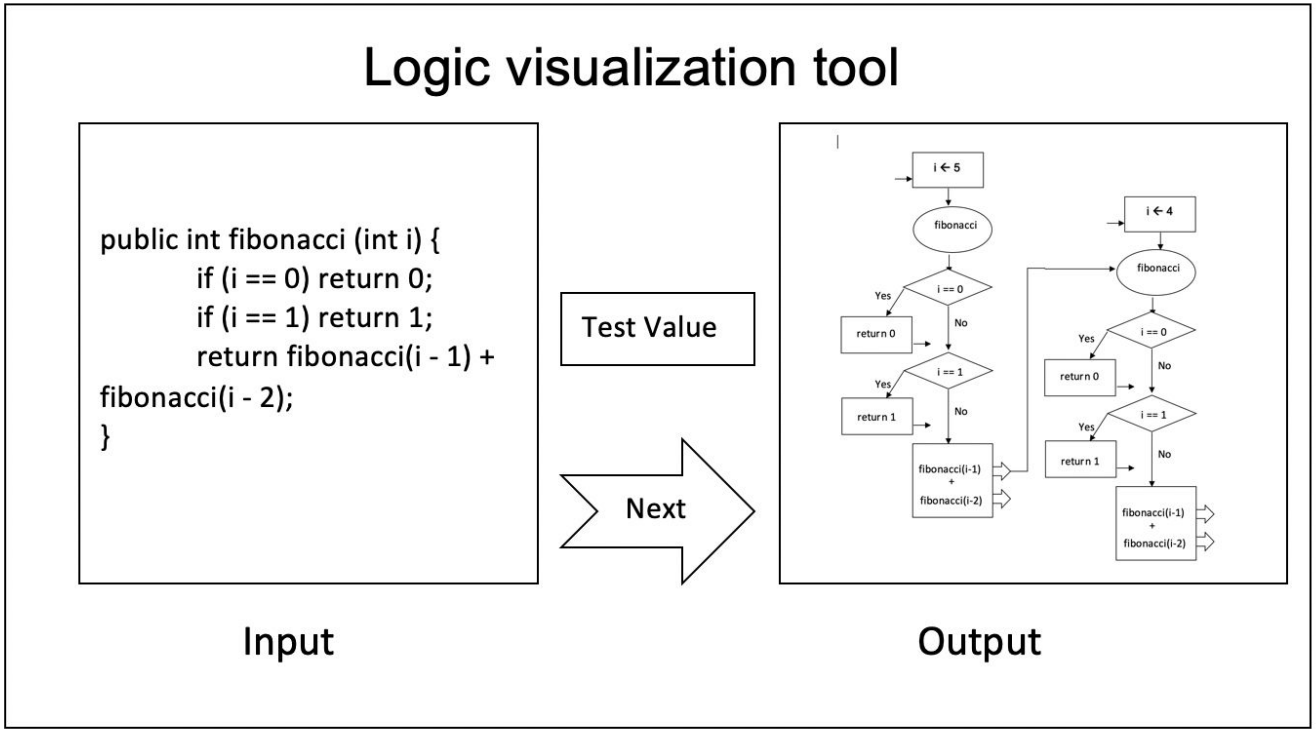


Example

```
public int fibonacci (int i) {
        if (i == 0) return 0;
        if (i == 1) return 1;
        return fibonacci(i - 1) + fibonacci(i - 2);
}
```



Example


Logic visualization tool
public int fibonacci (int i) {
if (i == 0) return 0;
if (i == 1) return 1;
return fibonacci(i - 1) +
fibonacci(i - 2);
}
Test Value
Next
i ← 5
fibonacci
i == 0
Yes
return 0
No
i == 1
Yes
return 1
No
fibonacci(i-1)
+
fibonacci(i-2)
i ← 4
fibonacci
i == 0
Yes
return 0
No
i == 1
Yes
return 1
No
fibonacci(i-1)
+
fibonacci(i-2)
Input
Output

Outline

- Our Solution
- **Research**
- Challenges and Risks

Hypothesis

- By expanding the condensed form recursion usually takes, we can also unpack the difficulty of understanding the abstraction
- A tool to expand recursive code can help students understand the impact of every line they write

Metrics

- Focus on user studies
- Paper Prototyping set functions
- Measure Time and Understanding
- The actual effect of the tool on helping students write code is difficult to observe without the final product

Preliminary Results

- Learned to specify (wish we had known a while ago)
- Users like customization

Outline

- Our Solution
- Research
- **Challenges and Risks**

Challenges and Risks

1. Making the recursion visualization intuitive for everyone is difficult.
2. The scope of the project is not able to be decided due to the limit of time
    a. Integration with UI Frameworks
    b. Tracking of values during execution
    c. UI design

Conclusion

- The pursuit to teach Recursion better is not a novel idea
- Aim at the root of the problem in order to solve it

Questions?