CS 211: Recursion vs. Iteration

Chris Kauffman

Week 11-2
Front Matter

Goals Today

▶ Recursion

Lab 11: Task

P5: Gate Simulator

▶ Simulate small digital logic gates
▶ Medium sized project

Reading

▶ BJP Ch 12: Recursion
▶ Lab Manual Ch 18: Recursion

Schedule

4/6       W11       Recursion
4/13      W12       Review/Exam 2
4/20      W13       Linear/Binary Search
P5: Gate Simulator

- Mostly straight-forward/boiler-plate
- Largest Work is in Circuit
  - Recursive constructor (!)
  - Lots of stuff to hook together
- My current solution
  - 5hrs in
    - Tests run: 190, Failures: 0
- A few bugs/clarifications will be posted yesterday
- Good starting place: Prof Carver on @1284

Questions?
Recursion is...

Something specified in terms of a smaller version of itself
Recursion involves

**Base Case**
The "smallest thing", where you can definitively say "here is the answer"

**Inductive/Recursive Case**
If I had the answer to a few smaller versions of this problem, I could combine them to get the answer to this problem.
Identify Base and Recursive Cases

Fibonacci

\[ f_0 = 0 \]
\[ f_1 = 1 \]
\[ f_i = f_{i-1} + f_{i-2} \]

Factorial

\[ \text{fact}(n) = n \ast \text{fact}(n) \]
\[ \text{fact}(0) = 1 \]
Examine Stack Trace for Fibonacci

Recursive

public static int fibR(int n)

▶ Recursive implementation
▶ View Stack Trace of fibR(4)

Recursive

public static int fibI(int n)

▶ Iterative implementation?
▶ View Stack Trace of fibI(4)

Point
Recursion utilizes the Stack to store information about history
Other Uses for Recursion

Enumeration
Show me all possibilities of something
- All permutations of the numbers 1 to 10
- Print all games of Party Pong (hard problem from last year)

Search Problems
Show me whether something exists and how its put together
- Does a number exist in an array?
- Does a path exist from point z to point e on a grid and what is it?

```
| z . . . . . . . . . . . e |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
```
Exercise: Sums

- Print all permutations of positive numbers which total 8 (order of numbers matters)
- Create a recursive helper called `totalsTarget()`
- Base and recursive cases?

Prototypes

```java
public static void sumsTo8(){..}
public static void totalsTarget(int target,
                                 int current,
                                 String history)
```

target: Eight!
current: current total
history: numbers used so far

Example output

```
> javac Sums.java
> java Sums
8 = 1 1 1 1 1 1 1 1
8 = 1 1 1 1 1 1 2
8 = 1 1 1 1 1 2 1
8 = 1 1 1 1 1 3
8 = 1 1 1 1 2 1 1
8 = 1 1 1 1 2 2
8 = 1 1 1 1 3 1
8 = 1 1 1 1 3 2
8 = 1 1 1 2 1 1
8 = 1 1 1 2 2 1
8 = 1 1 1 3 1 1
8 = 1 1 1 3 2 1
8 = 1 1 2 1 1
8 = 1 1 2 2 1
8 = 1 2 1 1
8 = 2 1 1
8 = 6 1
8 = 6 2
8 = 7 1
8 = 8
```

- 128 lines...
- Iterative version?
Recall: Rabbits

A puzzle.¹

Consider the growth of an idealized (biologically unrealistic) rabbit population, assuming that:

- A newly born pair of rabbits, one male, one female, are put on an island;
- Rabbits are able to mate at the age of one month so that at the end of its second month a female can produce another pair of rabbits;
- Rabbits never die and a mating pair always produces one new pair (one male, one female) every month from the second month on.

How many pairs will there be in one year?

¹Adapted from Wikipedia
Tabularly

Mature pair produce baby pair the following month

**BN**  Baby pair from pair $N$

**MN**  Mature pair from pair $N$

<table>
<thead>
<tr>
<th>Month</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairs</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>Pair 0</td>
<td>B1</td>
<td>MI</td>
<td>MI</td>
<td>MI</td>
<td>MI</td>
<td>MI</td>
<td>MI</td>
<td>MI</td>
</tr>
<tr>
<td>Pair 1</td>
<td>B0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>Pair 2</td>
<td>B0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>Pair 3</td>
<td>B0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>Pair 4</td>
<td>B1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
</tr>
<tr>
<td>Pair 5</td>
<td>B0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>Pair 6</td>
<td>B1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
</tr>
<tr>
<td>Pair 7</td>
<td>B2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>Pair 8</td>
<td>B0</td>
<td>B0</td>
<td>B0</td>
<td>B0</td>
<td>B0</td>
<td>B0</td>
<td>B0</td>
<td>B0</td>
</tr>
<tr>
<td>Pair 9</td>
<td>B1</td>
<td>B1</td>
<td>B1</td>
<td>B1</td>
<td>B1</td>
<td>B1</td>
<td>B1</td>
<td>B1</td>
</tr>
<tr>
<td>Pair 11</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
<td>B3</td>
</tr>
<tr>
<td>Pair 12</td>
<td>B4</td>
<td>B4</td>
<td>B4</td>
<td>B4</td>
<td>B4</td>
<td>B4</td>
<td>B4</td>
<td>B4</td>
</tr>
</tbody>
</table>
Simulate

Model the generations of rabbits directly.

```java
public class RabbitSim{
    // Recursive!
    public static ArrayList<Rabbits> generation(int g);
    public static void main(String args[]);
}
```

```
lila [%w14-2-code]% java RabbitSim 1
  1 1 [BI]
lila [%w14-2-code]% java RabbitSim 3
  1 1 [BI]
  2 1 [MI]
  3 2 [MI, B0]
lila [%w14-2-code]% java RabbitSim 7
  1 1 [BI]
  2 1 [MI]
  3 2 [MI, B0]
  4 3 [MI, M0, B0]
  5 5 [MI, M0, M0, B0, B1]
  6 8 [MI, M0, M0, M0, M1, B0, B1, B2]
  7 13 [MI, M0, M0, M0, M1, M0, M1, M2, B0, B1, B2, B3, B4]
```

Discuss connection to an iterative solution