Recursive Tracing

Hitesh Boinpally
Summer 2023

Agenda

- Recursion Intro
- Practice
- Visualizations
- Reminders

Agenda

- Recursion Intro 
- Practice
- Visualizations
- Reminders

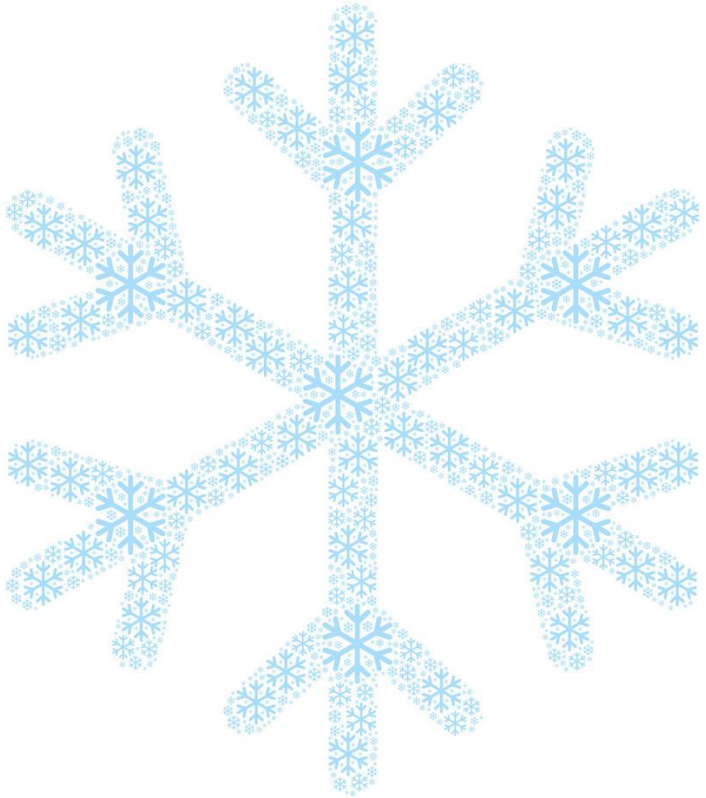
Road Map - Recursion

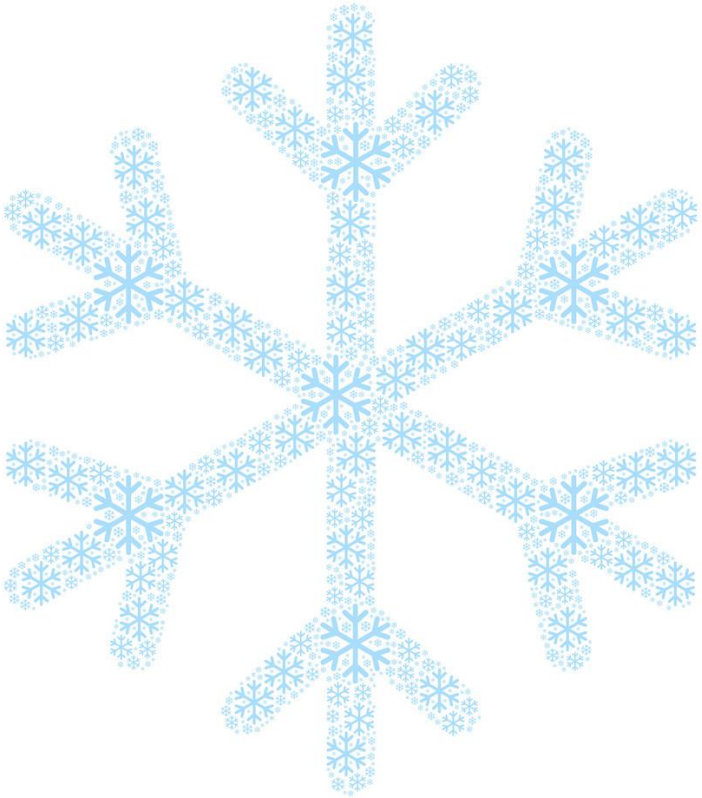
- Friday (Today)
  - Introduce idea of “recursion”
  - Goal: Understand idea of recursion and read recursive code
- Tuesday
  - Practice reading recursive code
- Wednesday
  - More complex recursive examples
  - Goal: Identify recursive structure in problems and write recursive code
- Thursday
  - Practice writing recursive code

Road Map - Recursion

- Friday (Today)
  - Introduce idea of “recursion”
  - Goal: Understand idea of recursion and read recursive code
- Tuesday
  - Practice reading recursive code
- Wednesday
  - More complex recursive examples
  - Goal: Identify recursive structure in problems and write recursive code
- Thursday
  - Practice writing recursive code

Recursion Intro





Recursion Intro

- **Recursion:** Defining something in terms of itself
- **Recursive Programming:** Writing methods that call themselves to solve problems
  - Helpful to solve specific problems
  - Equally as powerful as iterative solutions

Recursive Programming

Two cases to always keep in mind:

1. **Base Case:**
   - Stopping point, how to know we're "done"
   - Easiest / smallest thing to calculate
2. **Recursive Case:**
   - Do "one step" of the problem
   - Pass on the work to the next method call

- Some problems may have multiple base / recursive cases!

Agenda

- Recursion Intro
- Practice 
- Visualizations
- Reminders

Recursive Tracing Practice

slido.com
code: #su_cse123

```
public static int recursiveMystery(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * recursiveMystery(n - 1);
    }
}
```

Recursive Tracing Practice

slido.com
code: #su_cse123

```
public static int recursiveMystery(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * recursiveMystery(n - 1);
    }
}
```

Think of some example calls.
What do they execute?

`recursiveMystery(0)`
`recursiveMystery(2)`
`recursiveMystery(4)`

Recursive Tracing Practice

slido.com
code: #su_cse123

```
public static int recursiveMystery(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * recursiveMystery(n - 1);
    }
}
```

recursiveMystery(4):

Agenda

- Recursion Intro
- Practice
- Visualizations 
- Reminders

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	
text	

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	"1. Suits"

Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	"1. Suits"

Contents of file:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	"2. Barry"

Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

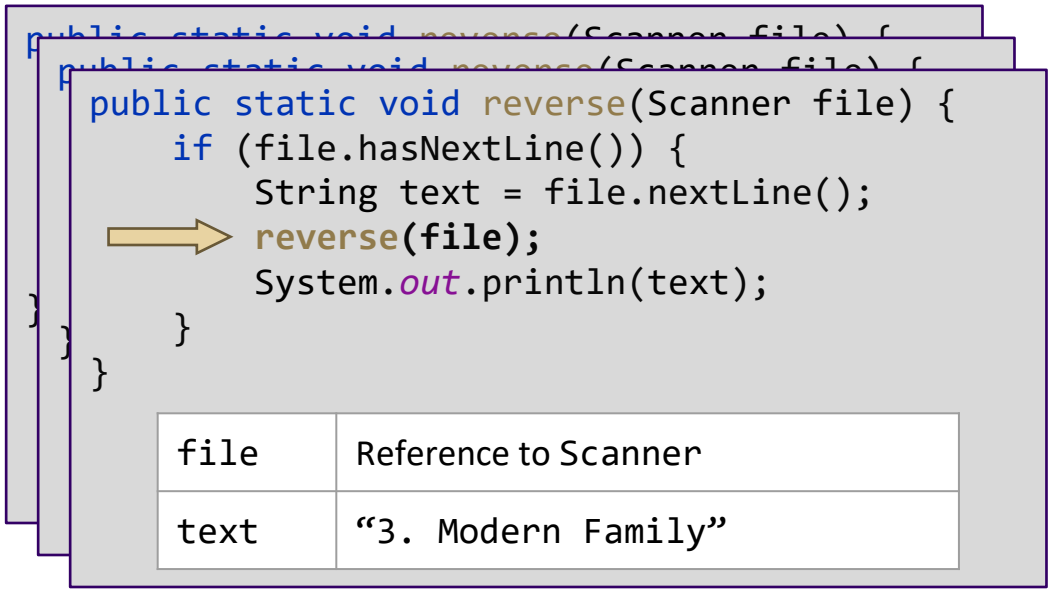
reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	"2. Barry"

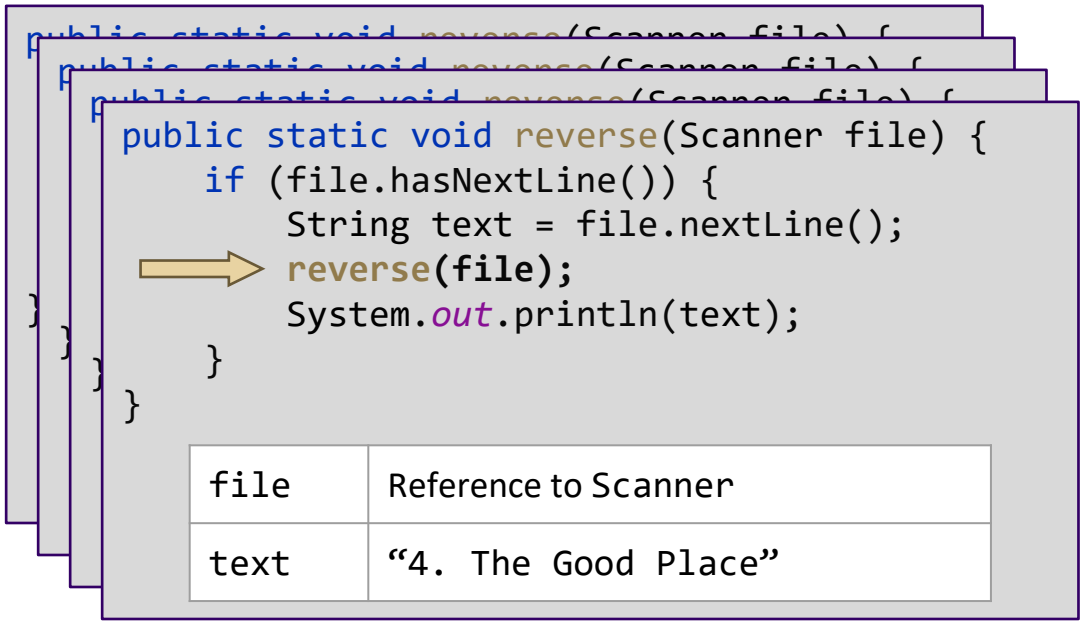
Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

reverse Visualization



Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

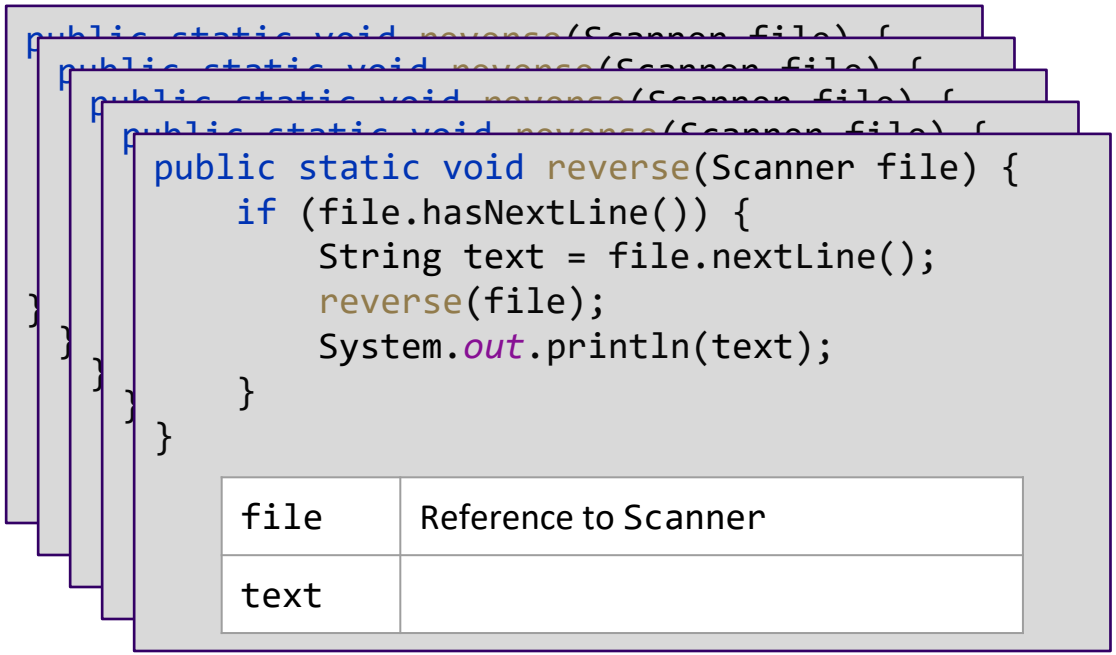
reverse Visualization



file	Reference to Scanner
text	"4. The Good Place"

Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

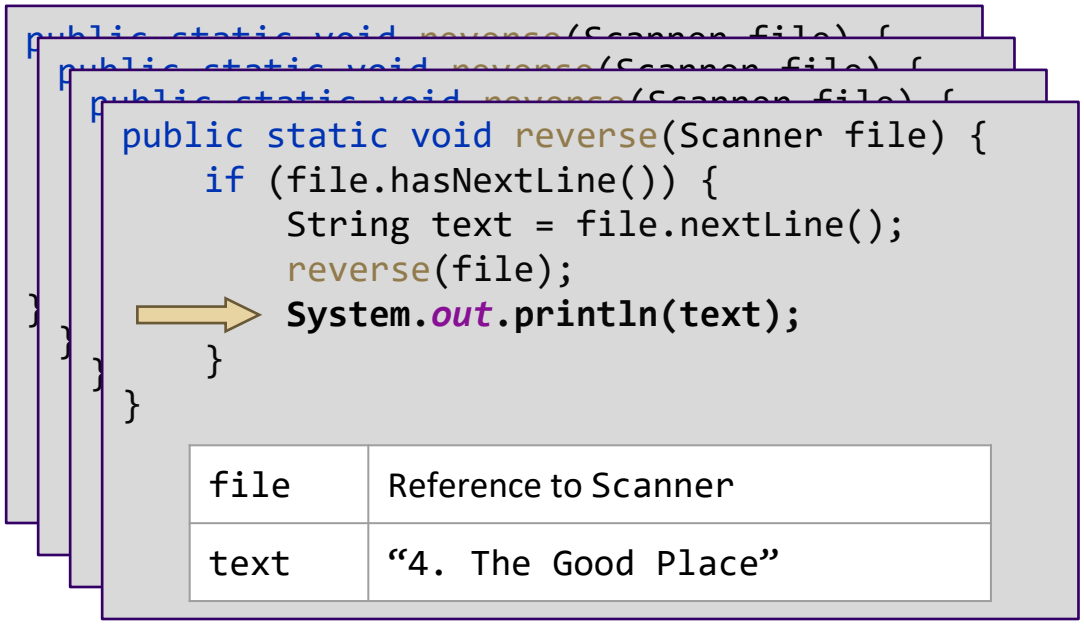
reverse Visualization



Contents of `file`:

1. Suits
2. Barry
3. Modern Family
4. The Good Place

reverse Visualization



file	Reference to Scanner
text	"4. The Good Place"

Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

Console output:
```
4. The Good Place
```

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	“3. Modern Family”

Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

Console output:
```
4. The Good Place
3. Modern Family
```

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	“2. Barry”

Contents of `file`:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

Console output:

```
4. The Good Place
3. Modern Family
2. Barry
```

reverse Visualization

```
public static void reverse(Scanner file) {
    if (file.hasNextLine()) {
        String text = file.nextLine();
        reverse(file);
        System.out.println(text);
    }
}
```

file	Reference to Scanner
text	"1. Suits"

Contents of file:
1. Suits
2. Barry
3. Modern Family
4. The Good Place

Console output:
```
4. The Good Place
3. Modern Family
2. Barry
1. Suits
```

Agenda

- Recursion Intro
- Practice
- Visualizations
- Reminders 

Reminders

- Resub 0 Due tonight
  - Updated to allow for C0 submissions

Reminders

- Resub 0 Due tonight
  - Updated to allow for C0 submissions
- Quiz 0 **Monday (7/10)**
  - Logistic details to be posted today
  - Topics: Linked Nodes, Linked Lists
  - Take-home, open 8:30am – 11:59pm
  - Will get instant feedback
  - Open collaboration (with a caveat), Open note
  - Reach out ASAP with any extenuating circumstances