# CSCI 136: Data Structures and Advanced Programming

## Lecture 8

## Recursion, part 2

Instructor: Dan Barowy

Williams

# Announcements

- Lab 3 d'oh!: ignore Github emails.
- Lab 1 feedback sent as pull request.
- Lab 3: partners assigned

# Outline

Recursion activity

Mathematical induction

# Activity

Write a method

```
public static int[] deriv(int[] poly)
```

that **computes the derivative** for a polynomial.

`poly` is an `int[]` that represents the **coefficients** of a polynomial. E.g.,

$$4x^3 + x^2 - 3x + 2$$

is represented by the array:

| | | | |
|---|---|---|---|
| 2 | -3 | 1 | 4 |
| 0 | 1 | 2 | 3 |

And `deriv` returns the array:

| | | |
|---|---|---|
| -3 | 2 | 12 |
| 0 | 1 | 2 |

which represents the coefficients of the **derivative**:

$$12x^2 + 2x - 3$$

# Activity Solution

```java
import java.util.Arrays;

class Derivative {
    public static int[] derivHelper(int[] poly, int i, int[] output) {
        if (i == 0) {
            return derivHelper(poly, i + 1, output);
        } else if (i == poly.length) {
            return output;
        }
        output[i - 1] = poly[i] * i;
        return derivHelper(poly, i + 1, output);
    }

    public static int[] deriv(int[] poly) {
        int[] output = new int[poly.length - 1];
        return derivHelper(poly, 0, output);
    }

    public static void main(String[] args) {
        int[] input = {2, -3, 1, 4};
        int[] output = deriv(input);
        System.out.println("input: " + Arrays.toString(input));
        System.out.println("output: " + Arrays.toString(output));
    }
}
```

# Mathematical Induction

# A note about "formal methods"

If the problem "fits" the mold, there is
a procedure for determining truth.

# Mathematical Induction

- The **mathematical cousin** of **recursion** is **induction**
- Induction is a **proof technique**
- Purpose: to **simultaneously prove** an **infinite number** of theorems!

## Principle of Mathematical Induction

Let **P(n)** be a **predicate** that is defined for **integers n**, and let **a** be a **fixed integer**.

**If** the following two statements are **true**:

1. **P(a)** is **true**.
2. For all integers **k ≥ a**, **if P(k)** is **true then P(k + 1)** is **true**.

**then** the statement

for all integers **n ≥ a**, **P(n)** is **true**

is **also true**.

## Principle of Mathematical Induction (variant)

Let **P(n)** be a **predicate** that is defined for **integers n**, and let **a** be a **fixed integer**.

**If** the following two statements are **true**:

1. **P(a)** is **true**.
2. For all integers **k > a**, **if P(k-1)** is **true then P(k)** is **true**.

**then** the statement

for all integers **n ≥ a**, **P(n)** is **true**

is **also true**.

## To be clear:

If you want to prove that **P(n)** is **true** for all integers **n ≥ a,**

1. You must first prove that **P(a)** is **true**.

2. Then you must prove that:

For all integers **k ≥ a**, **if P(k)** is **true then P(k+1)** is **true**.

**Critically**, when proving #2, **assume** that **P(k)** is **true** and **show** that **P(k+1) must also be true**.

## Names

Hypothesis: **P(n)** is **true** for all integers **n ≥ a,**

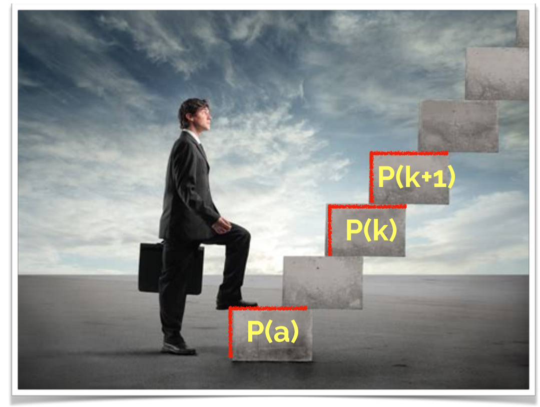
1. Base case: **P(a)** is **true**.

2. Inductive step:

For all integers **k ≥ a**, **if P(k)** is **true then P(k+1)** is **true**.

## Like recursion, there is an analogy

## Like recursion, there is an analogy




## Example

Prove that the sum of the first n integers is:

$$\frac{n(n+1)}{2}$$

## Recap & Next Class

### Today we learned:

Recursion (more)

Mathematical induction

### Next class:

More mathematical induction!