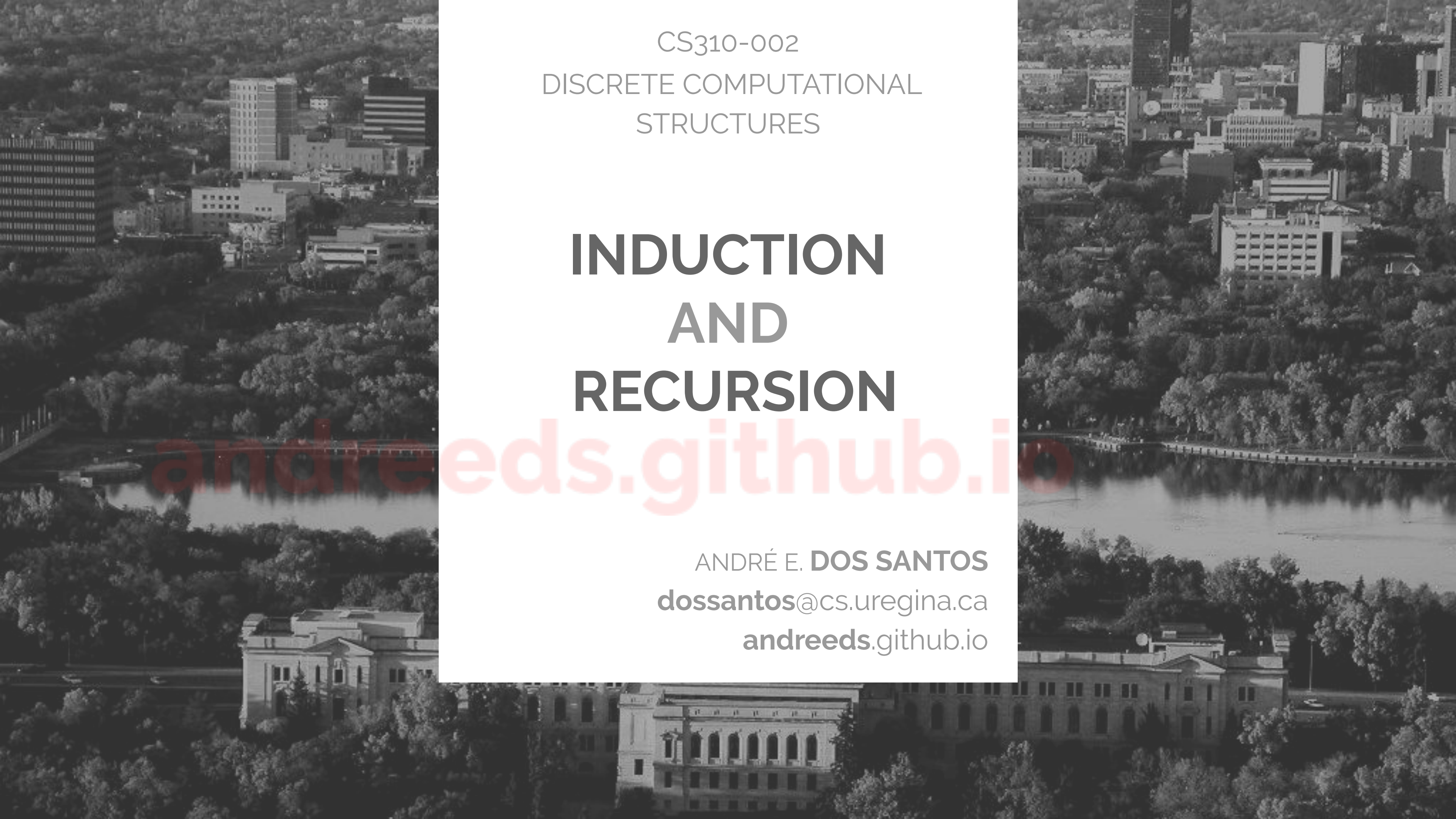
UNIVERSITY OF REGINA

CS310-002
DISCRETE COMPUTATIONAL STRUCTURES

ANDRÉ E. DOS SANTOS
dossantos@cs.uregina.ca
andreeds.github.io



CS310-002

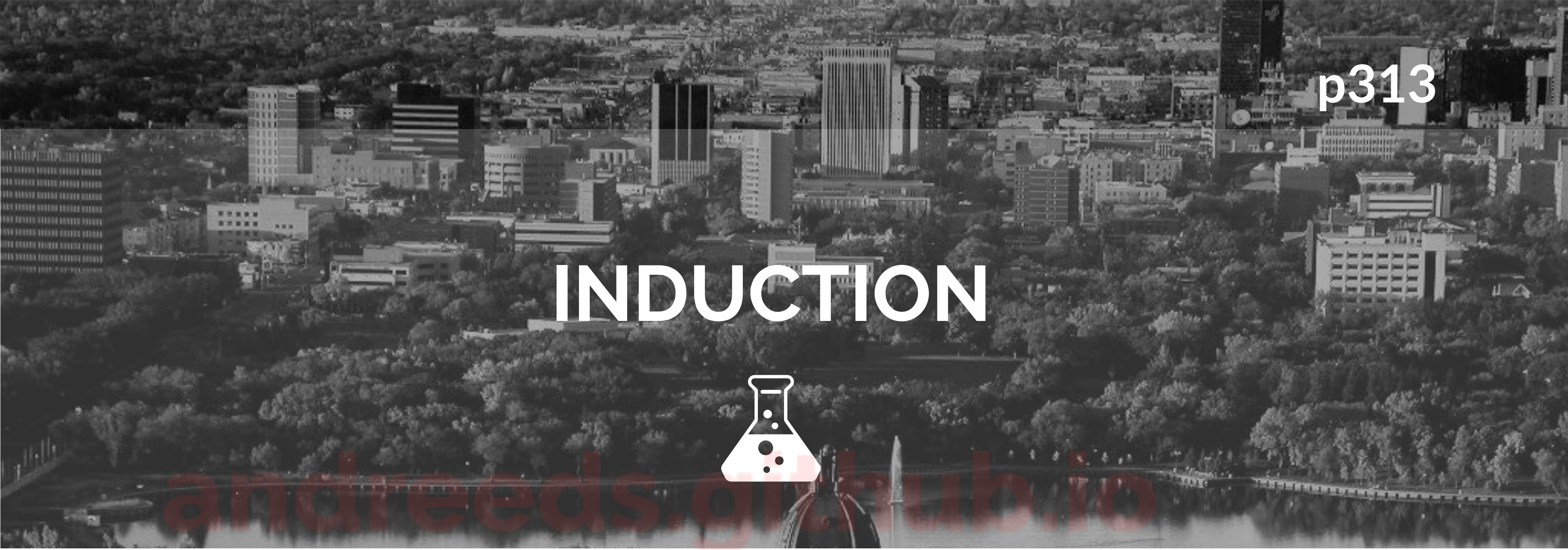
DISCRETE COMPUTATIONAL STRUCTURES

INDUCTION AND RECURSION

ANDRÉ E. **DOS SANTOS**

dossantos@cs.uregina.ca

andreeds.github.io


p313
INDUCTION

PRINCIPLE OF MATHEMATICAL INDUCTION

To prove that **P(n)** is true for all positive integers **n**, where **P(n)** is a propositional function, we complete two steps:

i. **BASIS STEP**

We verify that **P(1)** is true

ii. **INDUCTIVE STEP**

We show that the conditional statement **P(k) → P(k + 1)** is true for all positive integers **k**

PRINCIPLE OF MATHEMATICAL INDUCTION

This proof technique can be stated as:

$$(P(1) \land \forall k(P(k) \to P(k+1))) \to \forall n P(n),$$

when the domain is the set of positive integers

Example:

Use mathematical induction to show that

$$1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Example:

Use mathematical induction to show that

$$1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$$

STRONG INDUCTION

PRINCIPLE OF STRONG INDUCTION

To prove that **P(n)** is true for all positive integers **n**, where **P(n)** is a propositional function, we complete two steps:

i. **BASIS STEP**

We verify that **P(1)** is true

ii. **INDUCTIVE STEP**

We show that the conditional statement $[\,P(1) \land P(2) \land \cdots \land P(k)\,] \rightarrow P(k+1)$ is true for all positive integers **k**

PRINCIPLE OF STRONG INDUCTION

This proof technique can be stated as:

$$P(1) \land \forall k([\, P(1) \land P(2) \land \cdots \land P(k)\,] \rightarrow P(k+1)) \rightarrow \forall n P(n),$$

when the domain is the set of positive integers

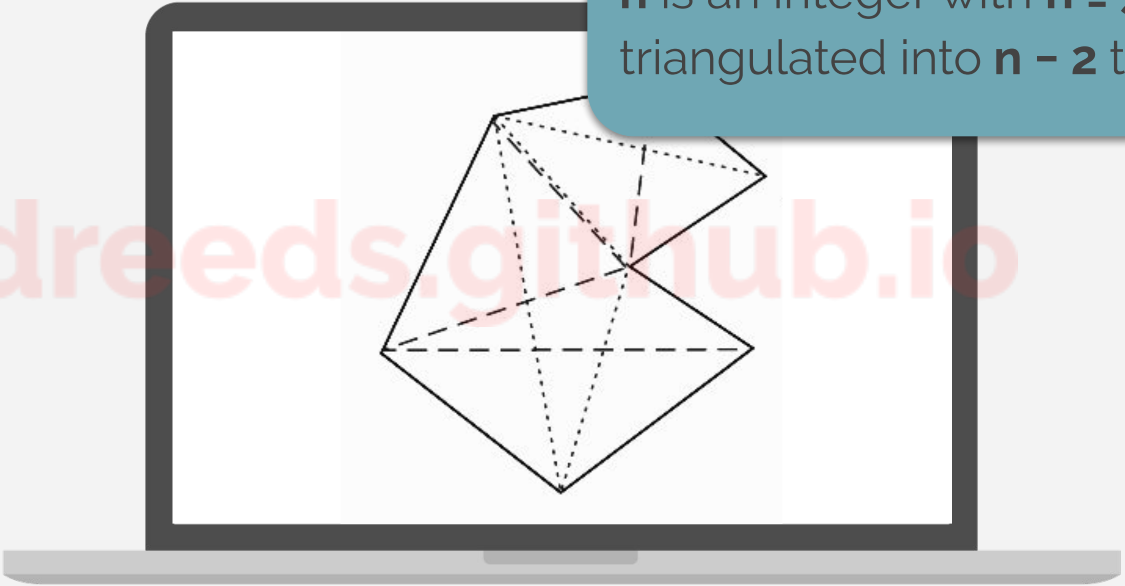
Example:

Show that if n is an integer greater than 1, then n can be written as the product of primes.

WORTH CHECKING OUT

p339

A simple polygon with **n** sides, where **n** is an integer with **n ≥ 3,** can be triangulated into **n − 2** triangles



RECURSIVE ALGORITHMS

PRINCIPLE OF MATHEMATICAL RECURSION

We use two steps to define a function with the set of nonnegative integers as its domain:

i. **BASIS STEP**

Specify the value of the function at zero.

ii. **RECURSIVE STEP**

Give a rule for finding its value at an integer from its values at smaller integers

Example:

Give a recursive definition of

$$\sum_{k=0}^{n} a_k.$$

Definition

An algorithm is called **recursive** if it solves a problem by reducing it to an instance of the same problem with smaller input

Example

Fibonacci numbers is a sequence of integers **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...** , which can be generated by the following recurrence equation:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \text{ for } n \geq 2$$

A ITERATIVE ALGORITHM FOR FIBONACCI NUMBERS

Input: A nonnegative integer n.
Output: The n-th Fibonacci number.

```
if n = 0 then
    return 0
else
    x ← 0 ;
    y ← 1 ;
    for i = 1 to n − 1 do
        z ← x + y ;
        x ← y ;
        y ← z ;
return y
```

A RECURSIVE ALGORITHM FOR FIBONACCI NUMBERS

Input: A nonnegative integer n.
Output: The n-th Fibonacci number.

Procedure *fibonacci(n: nonnegative integer)*
if $n = 0$ **then**
  return *0*
else if $n = 1$ **then**
  return *1*
else
  return *fibonacci(n-1)* + *fibonacci(n-2)*

Example:

Give a recursive algorithm for computing $n!$, where n is a nonnegative integer.

Example:

Give a recursive algorithm for computing the greatest common divisor of two nonnegative integers a and b with $a < b$.

REVIEW QUESTIONS

INDUCTION AND STRONG INDUCTION

Use mathematical induction to show that

$$1^2 + 2^2 + 3^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Use strong induction to show that if a simple polygon with at least four sides is triangulated, then at least two of the triangles in the triangulation have two sides that border the exterior of the polygon.

REVIEW QUESTIONS
RECURSION

Describe a recursive algorithm for multiplying two non-negative integers x and y based on the fact that $xy = 2(x \cdot (y/2))$ when y is even and $xy = 2(x \cdot \lfloor y/2 \rfloor) + x$ when y is odd, together with the initial condition $xy = 0$ when $y = 0$.

Give iterative and recursive algorithms for finding the nth term of the sequence defined by $a_0 = 1$, $a_1 = 3$, $a_2 = 5$, and $a_n = a_{n-1} \cdot a_{n-2}^2 \cdot a_{n-3}^3$. Which is more efficient?