MA/CSSE 473
Day 07

Mathematical Induction

Euclid's Algorithm

MA/CSSE 473 Day 07

- HW 3 is due tomorrow
- **Student Questions**
- The plan for after-the-fact homework discussions:
  - Led by students
  - Volunteers (until everyone has done 2).
- Mathematical induction review
  - Pie survivor
  - Tiling with Trominoes
- Euclid's algorithm

Induction Review

- To show that property P(n) is true for all integers $n \geq n_0$, it suffices to show:
  - **Ordinary Induction**
    - $P(n_0)$ is true
    - For all $k \geq n_0$, if P (k) is true, then P(k+1) is also true.

  or
  - **Strong Induction**
    - $P(n_0)$ is true
    - For all $k > n_0$, if P(j) is true for all j with $n_0 \leq j < k$, then P(k) is also true.

Proof by Induction

On Liquor Production by David M. Smith

A friend who's in liquor production
Owns a still of astounding construction.
The alcohol boils
Through old magnetic coils…
She says that it's "proof by induction."

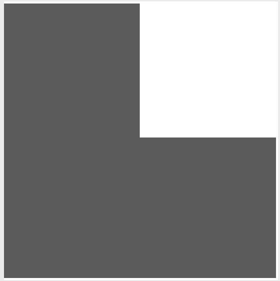
Disclaimer: The presentation of this multiple pun should not be taken as an implied endorsement on the part of the instructor of the production and/or consumption of liquor. For example, according to the National Institutes of Health (http://www.nih.gov/about/researchresultsforthepublic/AlcoholRelatedTrafficDeaths.pdf), 40% of traffic deaths involve alcohol. NIH studies revealed that young people who began drinking before age 15 are four times more likely to develop alcohol dependence during their lifetime than those who began drinking at age 21 or later. Those that drank before age 15 are also seven times more likely to report having been in a traffic crash because of drinking both during adolescence and adulthood. Alcohol also plays a significant role in risky sexual behavior and increases the risk of physical and sexual assault. Among college students under age 21, 50,000 experience alcohol-related date rape, and 43,000 are injured by another student who has been drinking. Each year, approximately 5,000 persons under the age of 21 die from causes related to underage drinking. These deaths include about 1,600 homicides and 300 suicides.

Another Induction Example

- Pie survivor
  - An odd number of people stand in various positions such that no two distances between people are equal
    - Each person has a pie
    - A whistle blows, and each person simultaneously and accurately throws his/her pie at the nearest neighbor
  - **Claim:** No matter how the people are arranged, at least one person does not get hit by a pie
  - Let P(n) denote the statement: "There is a survivor in every odd pie fight with 2n + 1 people"
  - Prove by induction that P(n) is true for all $n \geq 1$

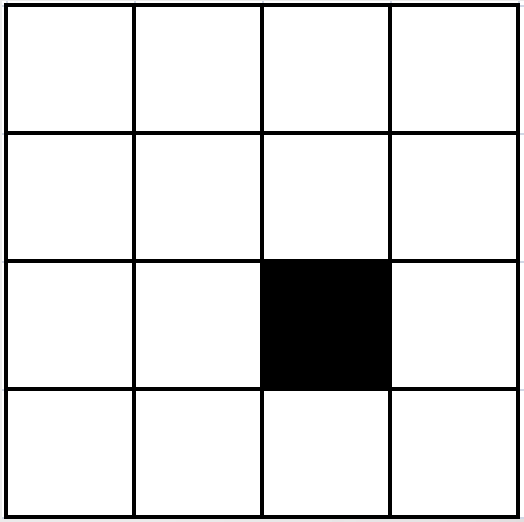
One More Induction Example

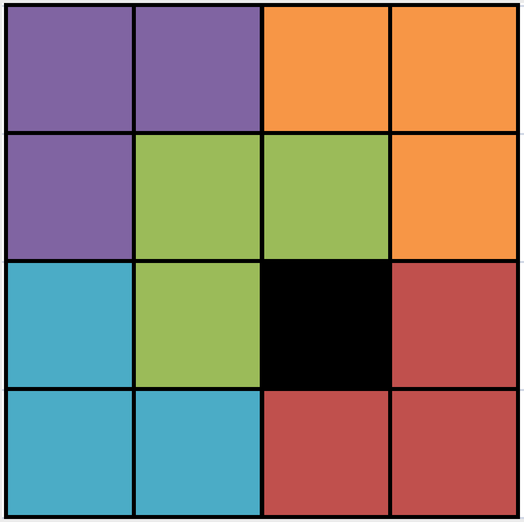
- Tiling with Trominoes
- We saw that a $2^n \times 2^n$ checkerboard can be tiled with dominoes.
- What about trominoes?
- Clearly, we can't tile an entire board!
- **Definition:** A **deficient** rectangular grid of squares is one that has one square missing.
- It's easy to see that we can tile a 2×2 deficient rectangle!



Trominoes Continued

- What about a 4 x 4 deficient rectangle?
- Can we tile this?





Trominoes Continued

- Prove by induction that we can tile any $2^n \times 2^n$ deficient rectangle with trominoes
- Base case: n=1   Done
- Assume that we can do it for n=k
- Show that we can do it for n=k+1
- Assume WLOG that the missing square is in the lower right quadrant of the rectangle
  - If it is somewhere else, we could simply rotate the board.

Structural Induction

- When a structure is defined recursively, use induction on the structural definition to prove that the property is true for everything covered by the definition
- Base case: The base cases in the recursive definition
- Induction step: Each recursive part of the definition
- We could express it as ordinary induction based on some metric of the structure, but it is often easier to do the induction on the structure itself.

Structural Induction Example 1

- Consider the following oversimplified definition of expressions in a programming language:
  - <Exp> ::= <number>
  - <Exp> ::= <Exp> <op> <Exp>
  - <Exp> ::= **(** <Exp> **)**
  - <op> ::= **+** | **-** | ***** | **/**
- Prove by structural induction: anything that can be derived from this grammar has an even number of parentheses

Structural Induction Example 2

- An Extended Binary Tree (EBT) **T** is either:
  - An external node, (designated by a square in diagrams), or
  - An internal node (designated by a circle in diagrams), and two subtrees (T_L and T_R) which are themselves EBTs. This internal node is called the **root** of the tree
- **Notation:** EN(**T**) and IN(**T**) denote the number of external nodes and internal nodes, respectively, in the Extended Binary Tree **T**.
- Prove by structural induction:
  In every EBT **T**, EN(**T**) = IN(**T**) + 1

Euclid's Algorithm: the problem

- One of the oldest known algorithms (about 2500 years old)
- **The problem:** Find the greatest common divisor (gcd) of two non-negative integers a and b.
- The approach you learned in grade school:
  - Completely factor each number
  - find common factors (with multiplicity)
  - multiply them all together to get the gcd
- Factoring is hard!
- Simpler approach needed

Euclid's Algorithm: the basis

- Based on the following rule:
  - If x and y are positive integers with $x \geq y$, then $\gcd(x, y) = \gcd(y, x \bmod y)$
- Proof of Euclid's rule:
  - It suffices to show the simpler rule

    $$\gcd(x, y) = \gcd(y, x - y)$$

    since x mod y can be obtained from x and y by repeated subtraction
  - Any integer that divides both x and y must also divide $x – y$, so $\gcd(x, y) \leq \gcd(y, x – y)$
  - Any integer that divides both x and $x - y$ must also divide y, so $\gcd(x, x-y) \leq \gcd(y, x)$

Euclid's Algorithm: the algorithm

```python
def euclid(a, b):
    """ INPUT:  Two integers a and b with a >= b >= 0
        OUTPUT: gcd(a, b)"""
    if b == 0:
        return a
    return euclid(b, a % b)
```

- Example: euclid(60, 36)
- Does the algorithm work?
- How efficient is it?