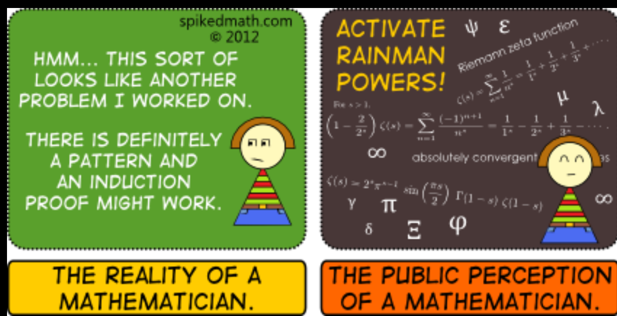



http://spikedmath.com/528.html

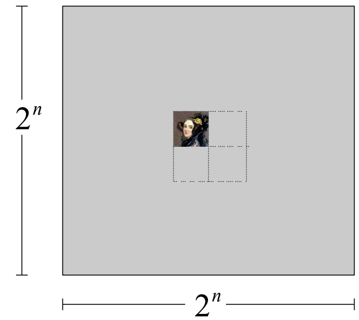
# CS151 Fall 2014
# Lecture 10 – 9/25
# Induction

Prof. Tanya Berger-Wolf
http://www.cs.uic.edu/CS151

Adapted from Lap Chi Lau – The Chinese University of Hong Kong
and David Liben-Nowell – Carleton College

## Puzzle

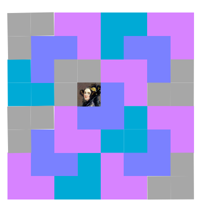
Goal: tile the squares, except one in the middle for Ada.




## Puzzle

There are only L-shaped tiles covering three squares:

For example, for 8 x 8 puzzle might tile for Ada this way:



## Puzzle

Theorem: For any $2^n$ x $2^n$ puzzle, there is a tiling with Ada in the middle.

Did you remember that we proved $2^{2n} - 1$ is divisible by 3?

Proof: (by induction on $n$)

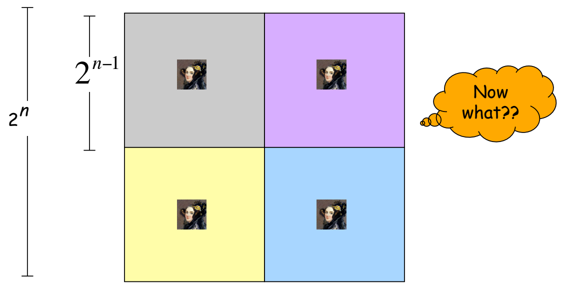
$P(n)$ ::= can tile $2^n$ x $2^n$ with Ada in middle.

Base case: ($n$=0)

(no tiles needed)

## Puzzle

Induction step: assume can tile $2^{n-1} \times 2^{n-1}$,
prove can handle $2^n \times 2^n$.




## Puzzle

**The new idea:**

A stronger property

Prove that we can always find a tiling with Ada anywhere.

Theorem B: For any $2^n \times 2^n$ plaza, there is a tiling with Ada anywhere.

Clearly Theorem B implies Theorem.

Theorem: For any $2^n \times 2^n$ plaza, there is a tiling with Ada in the middle.

## Puzzle

Theorem B: For any $2^n \times 2^n$ plaza, there is a tiling with Ada anywhere.

Proof: (by induction on $n$)

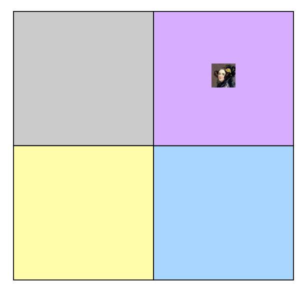
$P(n)$ ::= can tile $2^n \times 2^n$ with Ada anywhere.

Base case: ($n$=0)

(no tiles needed)

## Puzzle

Induction step:
*Assume* we can get Ada anywhere in $2^{n-1} \times 2^{n-1}$.
*Prove* we can get Ada anywhere in $2^n \times 2^n$.

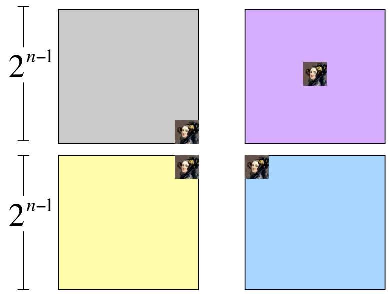


## Puzzle

Induction step:

*Assume* we can get Ada anywhere in $2^{n-1} \times 2^{n-1}$.

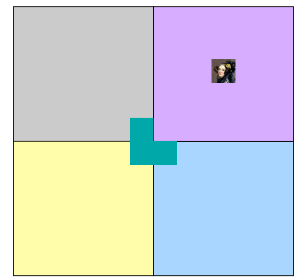
*Prove* we can get Ada anywhere in $2^n \times 2^n$.




## Puzzle

Method: Now group the squares together,

and fill the center with a tile.



**Done!**

## Ingenious Induction Hypothesis

**Note 1**: It may help to choose a *stronger hypothesis* than the desired result (e.g. "Ada in anywhere").

**Note 2**: The induction proof of "Ada in corner" implicitly defines a recursive procedure for finding corner tilings.

## Prime Products

*Claim*: Every integer > 1 is a product of primes.

*Proof:* (by strong induction)

- Base case is easy.
- Suppose the claim is true for all 2 <= i < n.
- Consider an integer n.
- If n is prime, then we are done.
- So $n = k \cdot m$ for integers $k$, $m$ where $n > k,m > 1$.
- Since $k,m$ smaller than n,
- *By the induction hypothesis, both k and m are product of primes*

$$k = p_1 \cdot p_2 \cdots p_{94}$$
$$m = q_1 \cdot q_2 \cdots q_{214}$$

## Prime Products

*Claim*: Every integer > 1 is a product of primes.

…So

$$n = k \cdot m = p_1 \cdot p_2 \cdots p_{94} \cdot q_1 \cdot q_2 \cdots q_{214}$$

is a prime product.

∴ This completes the proof of the induction step.

## Strong Induction

Strong induction

Prove $P(0)$.

Then prove $P(n)$ assuming *all* of

$P(0), P(1), \ldots, P(n\text{-}1)$ (instead of just $P(n\text{-}1)$).

Conclude **For all** $n.P(n)$

equivalent

Ordinary induction

$0 \rightarrow 1, 1 \rightarrow 2, 2 \rightarrow 3, \ldots, n\text{-}1 \rightarrow n.$

So by the time we got to $n$, already know *all* of

$P(0), P(1), \ldots, P(n\text{-}1)$

## Binary Search

I have a number between 0 and 63. You ask a question, I'll tell you yes or no.

How long will it take you to find my secret number?

```
BinarySearch (0…n-1)
middle = floor((n-1)/2)
if (middle == "secret number")
        return (middle)
else if (middle > "secret number")
                BinarySearch(0..middle)
    else
                BinarySearch(middle+1..n-1)
```