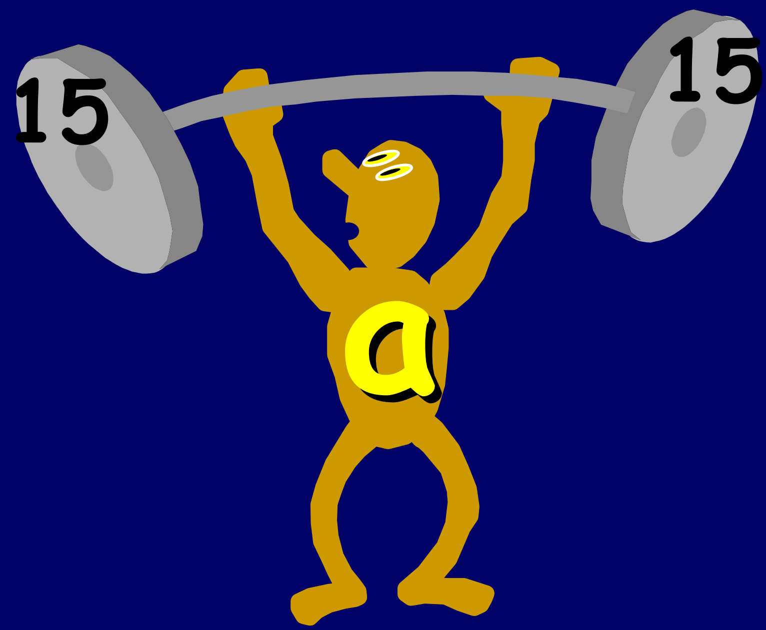
Great Theoretical Ideas In Computer Science

Anupam Gupta CS 15-251 Fall 2006

Lecture 5 Sept 12, 2006 Carnegie Mellon University

Ancient Wisdom: On Raising A Number To A Power



Egyptian Multiplication

The Egyptians used decimal numbers but multiplied and divided in binary

a x b By Repeated Doubling

b has n-bit representation: $b_{n-1}b_{n-2}\ldots b_1b_0$

Starting with a,
repeatedly double largest number so far
to obtain: $a, 2a, 4a, \ldots., 2^{n-1}a$

Sum together all the $2^k a$ where $b_k = 1$

$$b = b_0 2^0 + b_1 2^1 + b_2 2^2 + \ldots + b_{n-1} 2^{n-1}$$

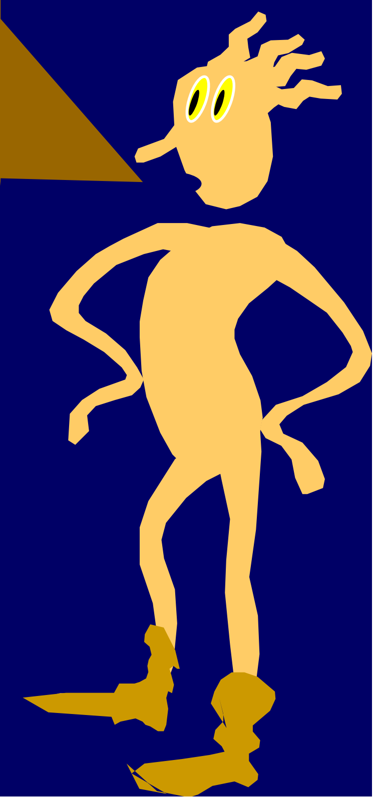
$$ab = b_0 2^0 a + b_1 2^1 a + b_2 2^2 a + \ldots + b_{n-1} 2^{n-1} a$$

$2^k a$ is in the sum if and only if $b_k = 1$

Wait!

How did the Egyptians do the part where they converted b to binary?

They used repeated halving to do base conversion!



Egyptian Base Conversion

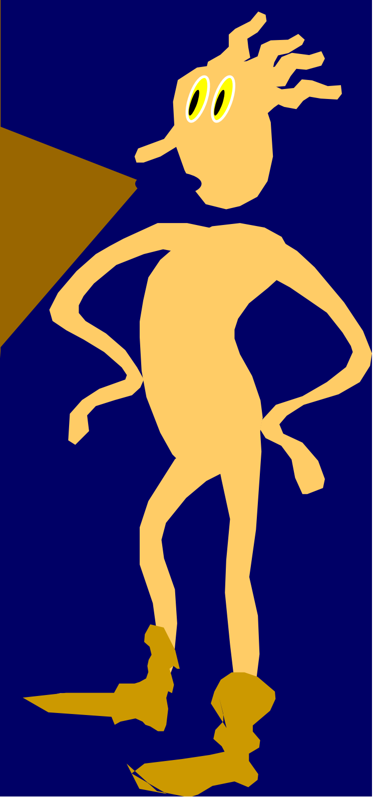
Output stream will print right to left

10 = 1010

```
Input X;
repeat {
    if (X is even)
       then print 0;
    else
       {X := X-1; print 1;}
    X := X/2 ;
} until X=0;
```

1010

Sometimes the Egyptians combined the base conversion by halving and multiplication by doubling into a single algorithm



70×13

Rhind Papyrus [1650 BC]

Binary for 13 is 1101 = $2^3 + 2^2 + 2^0$

$70*13 = 70*2^3 + 70*2^2 + 70*2^0$

Doubling	Halving	Odd?	Running Total
70	13	* 1	70
140	6	0	
280	3	* 1	350
560	1	* 1	910

$\Rightarrow 13 = (1101)_2$

30 x 5

Doubling	Halving	Odd?	Running Total
5	30	0	
10	15	* 1	10
20	7	* 1	30
40	3	* 1	70
80	1	* 1	150

184 / 17

Rhind Papyrus [1650 BC]

Doubling	Powers of 2	Check
17	1	
34	2	*
68	4	
$17 \times 8 =$ 136	8	*

$136 \times 2 > 184$

184 - 136 = 48 so check highest multiple of 17 less than 48

184 / 17

Rhind Papyrus [1650 BC]

Doubling	Powers of 2	Check
17	1	
34	2 ✓	*
68	4	
136	8 ✓	*

$136 \times 2 > 184$

48 - 34 = 14 so don't check anything

$\frac{184}{17} = 10$ remainder of 14.

184 / 17

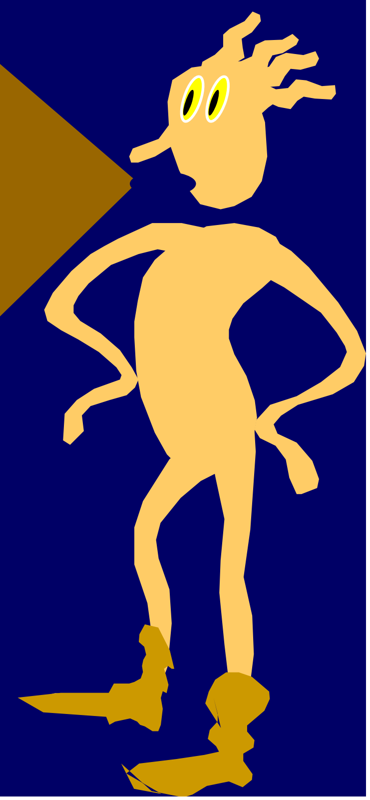
Rhind Papyrus [1650 BC]

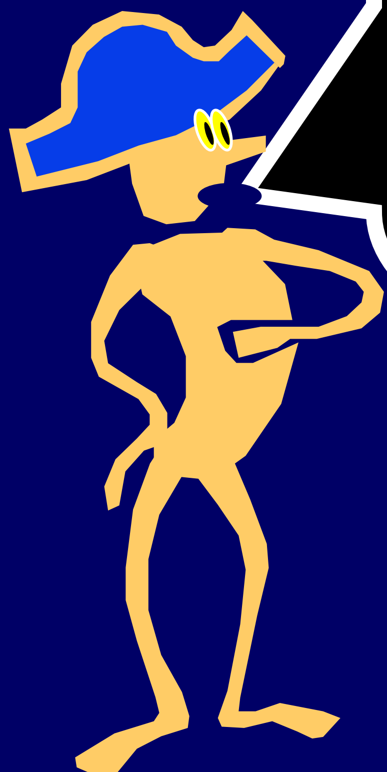
Doubling	Powers of 2	Check
17	1	
34	2	*
68	4	
136	8	*

$184 = 17*8 + 17*2 + 14$

$184/17 = 10$ with remainder 14

This method is called
"Egyptian Multiplication / Division"
or
"Russian Peasant Multiplication / Division"





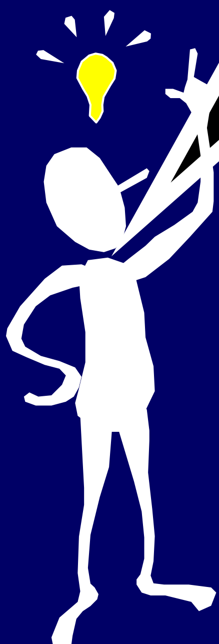
Wow! Those Russian peasants were pretty smart

Standard Binary Multiplication
=   Egyptian Multiplication

```
        * * * * * * * *
  ×                1101
  ---------------------
        * * * * * * * *

    * * * * * * * *
  * * * * * * * *
    ---------------------
* * * * * * * * * * * *
```

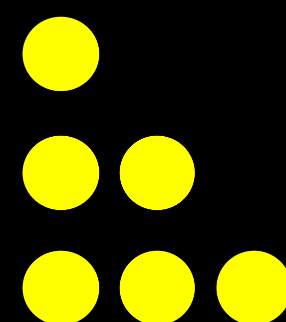
Our story so far…



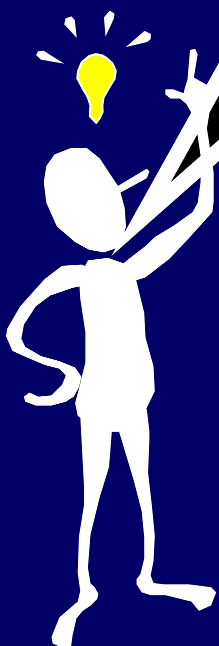
We can view numbers in many different, but corresponding ways

Representation:
Understand the relationship between different representations of the same information or idea

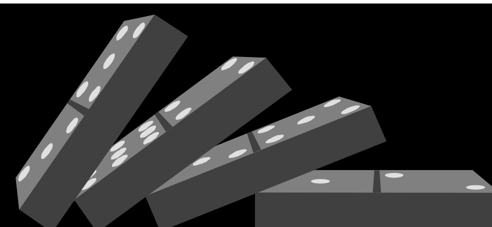
1
2
3



Our story so far…



Induction is how we define and manipulate mathematical ideas



Induction has many guises

Master their interrelationship

- Formal Arguments
- Loop Invariants
- Recursion
- Algorithm Design
- Recurrences

Let's Articulate New One:

Abstraction:

Abstract away the inessential features of a problem or solution

=

And it's "twin"

Generalization

Realize the inherent general power of problem or solution

=

Marcel Duchamp's L.H.O.O.Q

$b:=a^8$

```
b:=a*a
b:=b*a
b:=b*a
b:=b*a
b:=b*a
b:=b*a
b:=b*a
```

a^2 `b:=a*a`

a^4 `b:=b*b`

a^8 `b:=b*b`

This method costs only 3 multiplications. The savings are significant if $b:=a^8$ is executed often

Powering By Repeated Multiplication

Input: a,n

Output: Sequence starting with a, ending with a^n, such that each entry other than the first is the product of two previous entries

Example

Input: a,5

Output: a, a^2, a^3, a^4, a^5

$a \times a$ $\quad a^2 \times a$

or

Output: a, a^2, a^3, a^5

or

Output: a, a^2, a^4, a^5

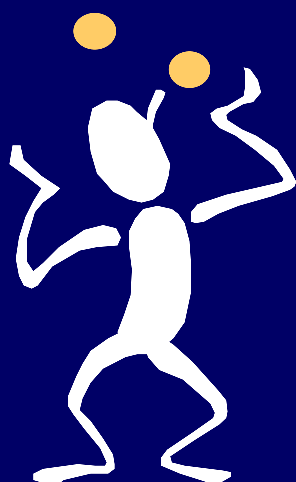
Given a constant n,
how do we implement
$b:=a^n$
with the fewest number
of multiplications?

Definition of M(n)

$M(n)$ = Minimum number of multiplications required to produce a^n from a by repeated multiplication

What is M(n)? Can we calculate it exactly? Can we approximate it?

Exemplification:
Try out a problem or solution on small examples




Very Small Examples

$M(n)$ = Minimum number of multiplications required to produce a^n from a by repeated multiplication

What is $M(1)$?

$M(1) = 0$ $[a]$

What is $M(0)$?

Not clear how to define $M(0)$

What is $M(2)$?

$M(2) = 1$ $[a, a^2]$

$M(8) = ?$

a, a^2, a^4, a^8 is one way to make a^8 in 3 multiplications

$a \quad a^2 \quad a^4 \quad a^8$

$M(8) \le 3$

What does this tell us about the value of $M(8)$?

$M(8) \le 3$

Upper Bound

$? \leq M(8) \leq 3$

$3 \leq M(8)$ by exhaustive search

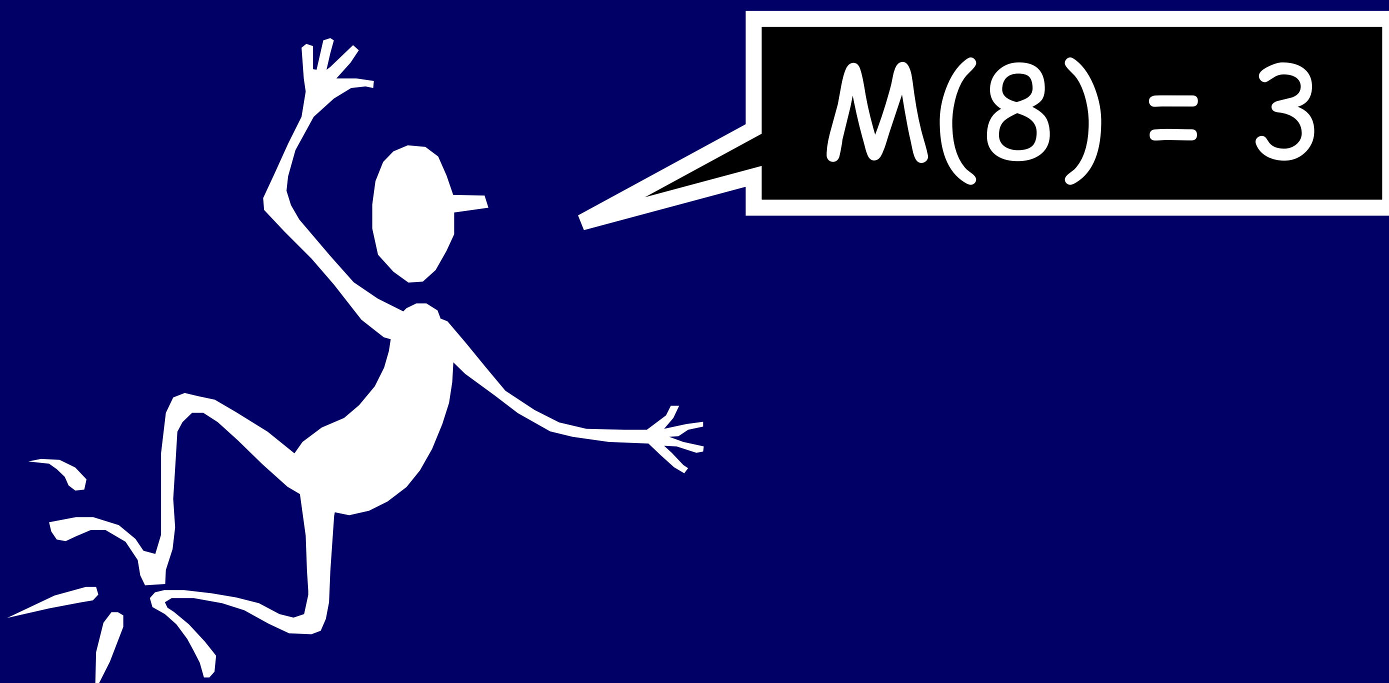
There are only two sequences with 2 multiplications. Neither of them make 8:

$$a, a^2, a^3 \quad \text{and} \quad a, a^2, a^4$$

$$3 \le M(8) \le 3$$

Lower Bound

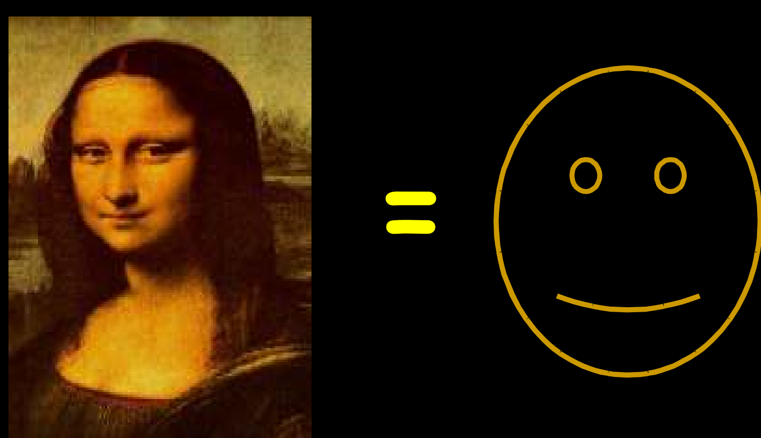
Upper Bound



What is the more essential representation of M(n)?

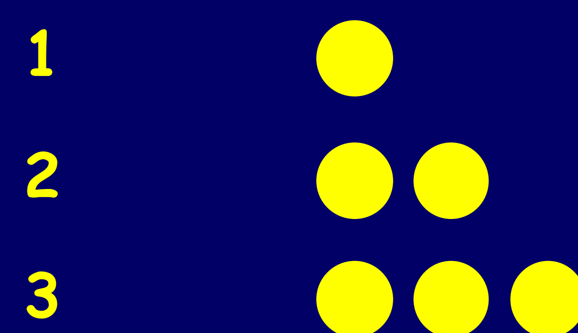
Abstraction

Abstract away the inessential features of a problem or solution



Representation:

Understand the relationship between different representations of the same idea



The "a" is a red herring

$a^x a^y$ is a^{x+y}

Everything besides the exponent is inessential. This should be viewed as a problem of repeated addition, rather than repeated multiplication

Addition Chains

$M(n) =$ Number of stages required to make n, where we start at 1 and in each stage we add two previously constructed numbers

Examples

Addition Chain for 8:

1 2 3 5 8

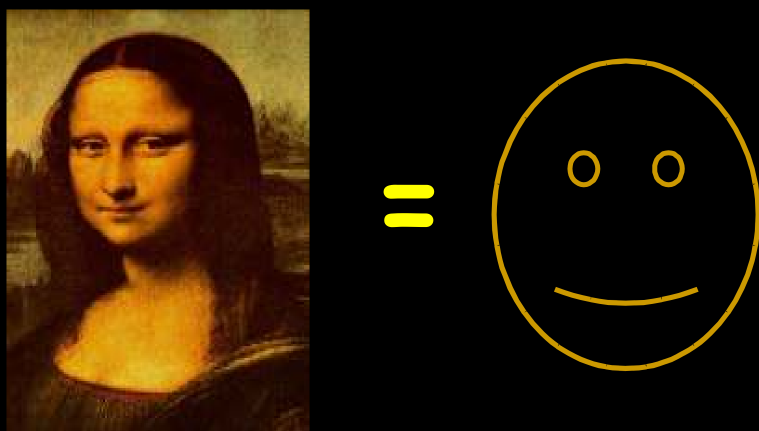
Minimal Addition Chain for 8:

1 2 4 8

Addition Chains Are a Simpler Way To Represent The Original Problem

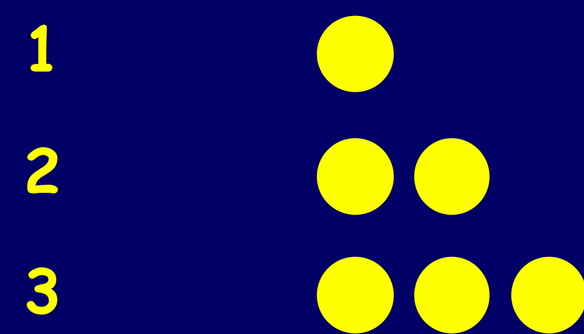
Abstraction

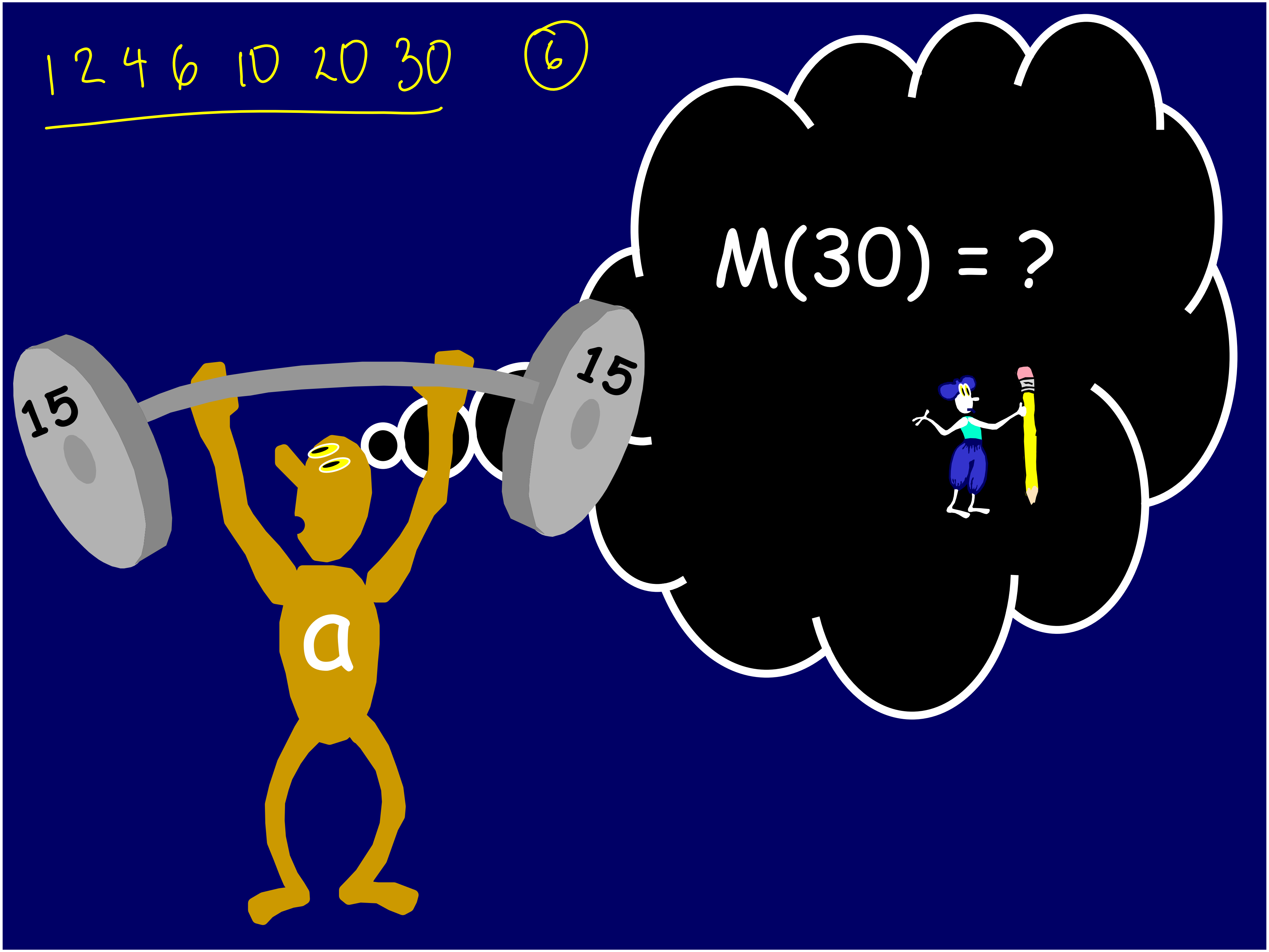
Abstract away the inessential features of a problem or solution



Representation:

Understand the relationship between different representations of the same idea





1 2 4 6 10 20 30 6
M(30) = ?
15
15
a

Addition Chains For 30

1	2	4	8	16	24	28	30
1	2	4	5	10	20	30	
1	2	3	5	10	15	30	
1	2	4	8	10	20	30	

$$? \le M(30) \le 6$$

$$? \le M(n) \le ?$$

100111

1 2 4 8 16 32

$32 + 4 = 36$

$36 + 2 = 38$

$38 + 1 = 39$

Binary Representation

Let B_n be the number of 1s in the binary representation of n

E.g.: $B_5 = 2$ since $5 = (101)_2$

Proposition: $B_n \leq \lfloor \log_2 (n) \rfloor + 1$

Proof: It is at most the number of bits in the binary representation of n

Binary Method

(Repeated Doubling Method)

Phase I (Repeated Doubling)

For $\lfloor \log_2 (n) \rfloor$ stages:

Add largest so far to itself

(1, 2, 4, 8, 16, . . .)

Phase II (Make n from bits and pieces)

Expand n in binary to see how n is the sum of B_n powers of 2. Use B_n - 1 stages to make n from the powers of 2 created in phase I

Total cost: $\lfloor \log_2 n \rfloor + B_n - 1$

Binary Method

Applied To 30

Phase I

1, 2, 4, 8, 16 Cost: 4 additions

Phase II

$30 = (11110)_2$

$2 + 4 = 6$

$6 + 8 = 14$

$14 + 16 = 30$ Cost: 3 additions

$$M(n) \le \lfloor \log_2 n \rfloor + B_n - 1 \le 2\lfloor \log_2 n \rfloor$$

Rhind Papyrus [1650 BC]

What is 30 x 5?

Addition chain for 30 →

1	5
2	10
4	20
8	40
16	80
24	120
28	140
30	150

← Start at 5 and perform same additions as chain for 30

Repeated doubling is the same as the Egyptian binary multiplication

Rhind Papyrus [1650 BC]

Actually used faster chain for 30*5

1	5
2	10
4	20
8	40
10	50
20	100
30	150

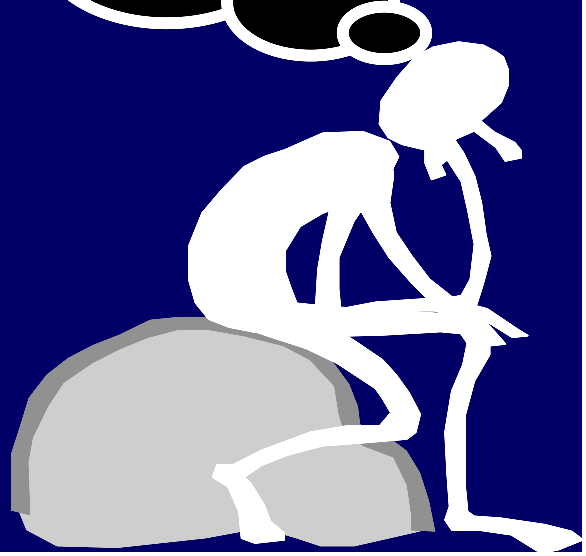
The Egyptian Connection

A shortest addition chain for n gives a shortest method for the Egyptian approach to multiplying by the number n

The fastest scribes would seek to know M(n) for commonly arising values of n

$$? \le M(30) \le 6$$

$$? \le M(n) \le 2 \lfloor \log_2 n \rfloor$$



A Lower Bound Idea

You can't make any number bigger than 2^n in n steps

1 2 4 8 16 32 64 . . .

Let S_k = "No k stage addition chain contains a number greater than 2^k"

Base case: k=0. S_0 is true since no chain can exceed 2^0 after 0 stages

$\forall\, k \geqslant 0, \quad S_k \Rightarrow S_{k+1}$

At stage k+1 we add two numbers from the previous stage

From S_k we know that they both are bounded by 2^k

Hence, their sum is bounded by 2^{k+1}.

Hence, no number greater than 2^{k+1} can be present by stage k+1

Change Of Variable

All numbers obtainable in m stages are bounded by 2^m. Let $m = \log_2(n)$

Thus, all numbers obtainable in $\log_2(n)$ stages are bounded by n

$$M(n) \geq \lceil \log_2 n \rceil$$

$5 \quad ? \le M(30) \le 6$

$$\lceil \log_2 n \rceil \le M(n) \le 2 \lfloor \log_2 n \rfloor$$

Theorem: 2^k is the largest number that can be made in k stages, and it can only be made by repeated doubling.

Proof by Induction:

Base $k = 0$ is clear

To make anything as big as 2^k requires having some number as big as 2^{k-1} in $k-1$ stages

By I.H., we must have all the powers of 2 up to 2^{k-1} at stage $k-1$. Hence, we can only double 2^{k-1} at stage k.

Theorem: M(30) > 5

Suppose that M(30)=5

At the last stage, we added two numbers x_1 and x_2 to get 30

Without loss of generality (WLOG), we assume that $x_1 \geq x_2$

Thus, $x_1 \geq 15$

By doubling bound, $x_1 \leq 16$

But $x_1 \neq 16$ since there is only one way to make 16 in 4 stages and it does not make 14 along the way. Thus, $x_1 = 15$ and M(15)=4

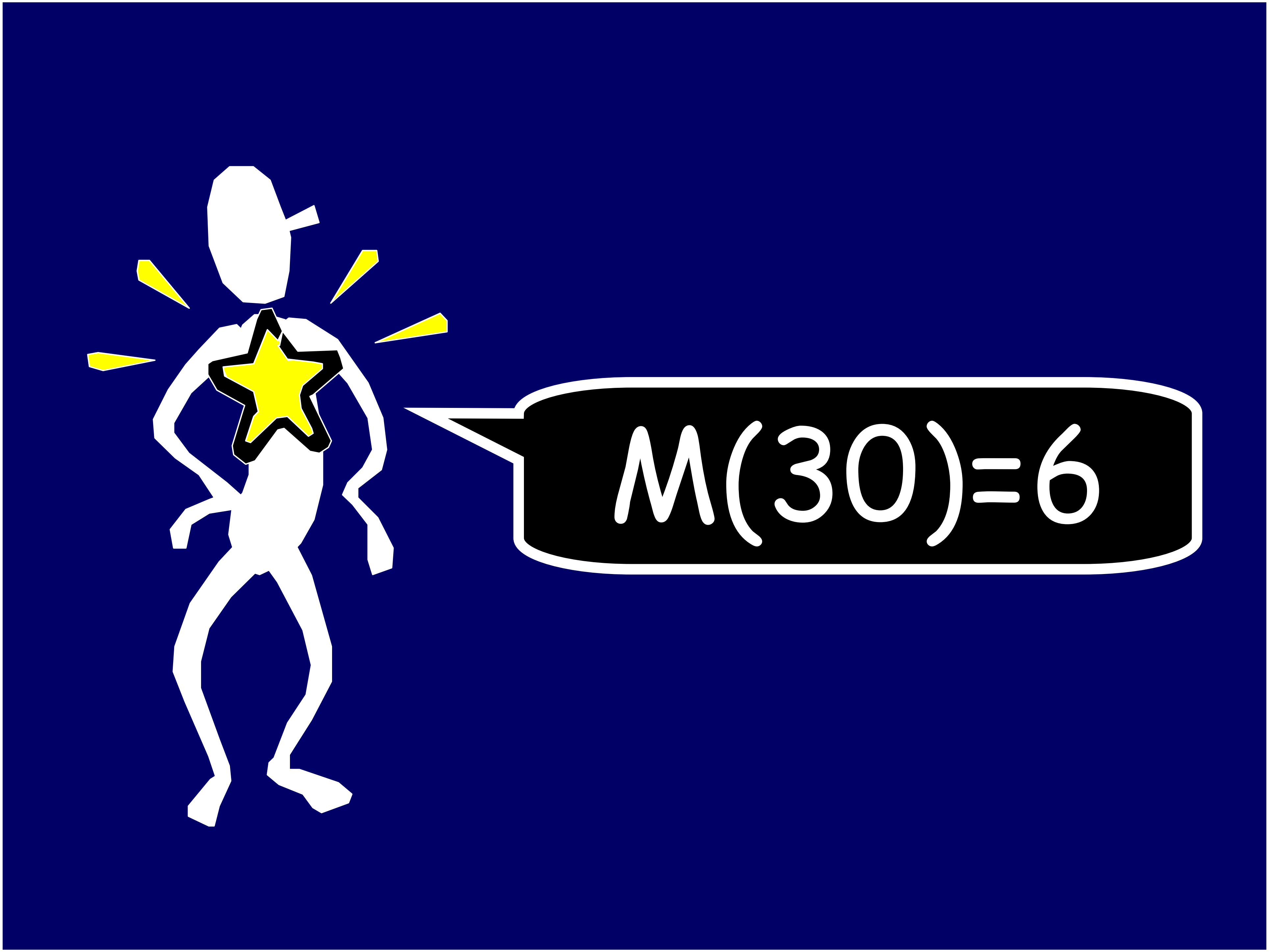
Suppose M(15) = 4

At stage 3, a number bigger than 7.5, but not more than 8 must have existed

There is only one sequence that gets 8 in 3 additions: 1 2 4 8

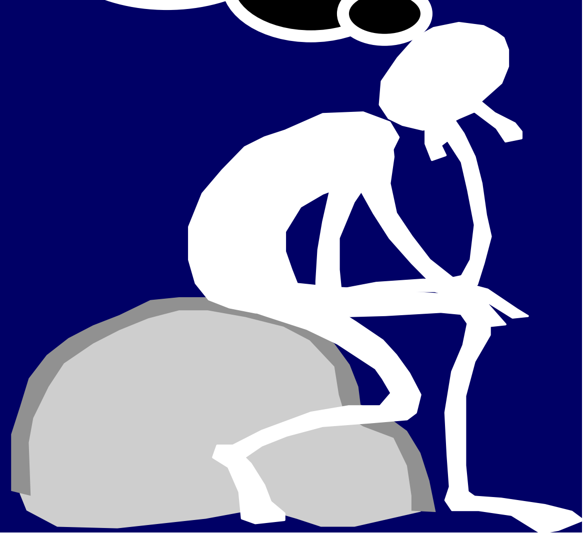
That sequence does not make 7 along the way and hence there is nothing to add to 8 to make 15 at the next stage

Thus, M(15) > 4   CONTRADICTION


M(30)=6

$$M(30) = 6$$

$$\lceil \log_2 n \rceil \le M(n) \le 2 \lfloor \log_2 n \rfloor$$



Better bounds??

Factoring Bound

$$M(a \times b) \leq M(a) + M(b)$$

Proof:

Construct a in M(a) additions

Using a as a unit follow a construction method for b using M(b) additions.
In other words, each time the construction of b refers to a number y, use the number ay instead

Example

$45 = 5 \times 9$

$M(5)=3$ [1 2 4 5]

$M(45) \leq 3 + 4$ [1 2 4 5 10 20 40 45]

"5 × [1 2 4 8 9]"

$M(9)=4$ [1 2 4 8 9]

Corollary (Using Induction)

$M(a_1a_2a_3\ldots a_n) \leq M(a_1)+M(a_2)+\ldots+M(a_n)$

Proof:

For n = 1 the bound clearly holds

Assume it has been shown for up to n-1

Now apply previous theorem using
$A = a_1a_2a_3\ldots a_{n-1}$ and $b = a_n$ to obtain:
$M(a_1a_2a_3\ldots a_n) \leq M(a_1a_2a_3\ldots a_{n-1}) + M(a_n)$

By inductive assumption,
$M(a_1a_2a_3\ldots a_{n-1}) \leq M(a_1) + M(a_2) + \ldots + M(a_{n-1})$

More Corollaries

Corollary: $M(a^k) \leq kM(a)$

Corollary: $M(p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_n^{\alpha_n}) \leq$
$\alpha_1 M(p_1) + \alpha_2 M(p_2) + \ldots + \alpha_n M(p_n)$

Does equality hold for $M(a \times b) \leq M(a) + M(b)$?

$M(33) < M(3) + M(11)$

$M(3) = 2$ [1 2 3]

$M(11) = 5$ [1 2 3 5 10 11]

$M(3) + M(11) = 7$

$M(33) = 6$ [1 2 4 8 16 32 33]

The conjecture of equality fails!

Conjecture: $M(2n) = M(n) + 1$

(A. Goulard)

A fastest way to an even number is to make half that number and then double it

Proof given in 1895 by E. de Jonquieres in L'Inter … vo…

FALSE! M(191)=M(382)=11

Furthermore, there are infinitely many such examples

Open Problem

Is there an n such that:

$M(2n) < M(n)$

Conjecture

Each stage might as well consist of adding the largest number so far to one of the other numbers

First Counter-example: 12,509
[1 2 4 8 16 17 32 64 128 256 512 1024 1041 2082 4164 8328 8345 12509]

Open Problem

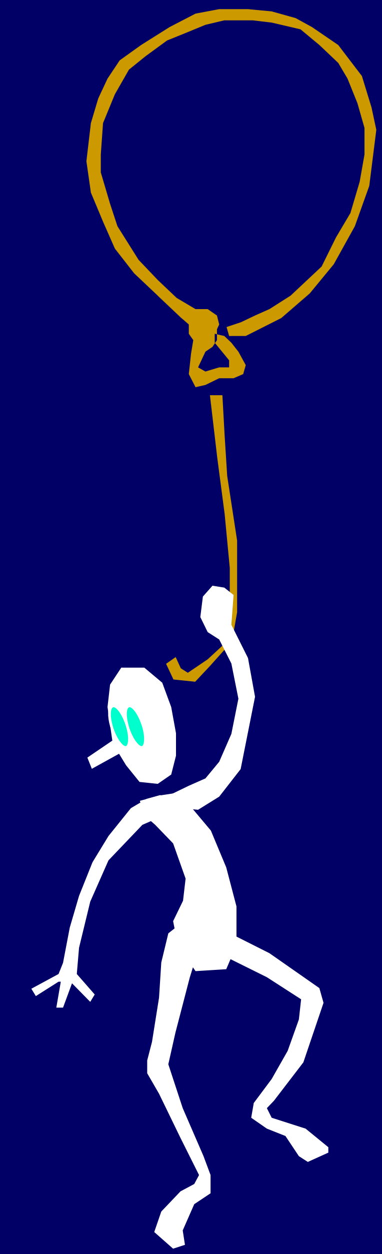
Prove or disprove the Scholz-Brauer Conjecture:

$$M(2^n-1) \leq n - 1 + B_n$$

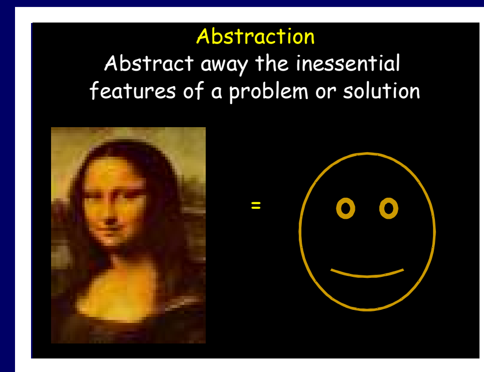
(The bound that follows from this lecture is too weak: $M(2^n-1) \leq 2n - 1$)

High Level Point

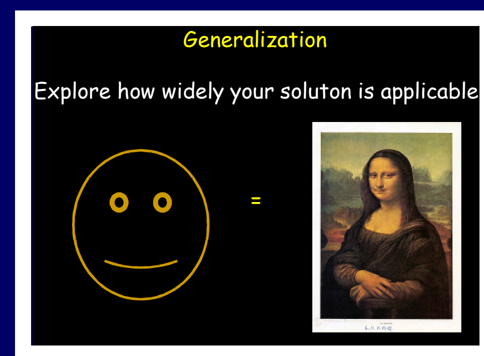
Don't underestimate "simple" problems. Some "simple" mysteries have endured for thousand of years



Raising a number to a power
with fewest multiplications



Shortest addition chain



Works for any structure which is
1. associative $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
2. a^k is well defined such that $a^{x+y} = a^x \bullet a^y$

E.g. adding and multiplying numbers
multiplying matrices, modular arithmetic

Study Bee

Egyptian Multiplication

Raising To A Power

Minimal Addition Chain

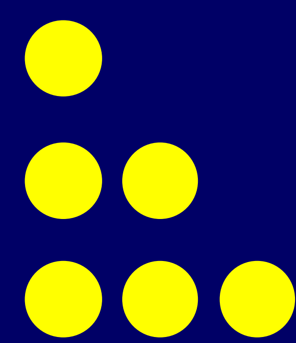
Lower and Upper Bounds

Repeated doubling method

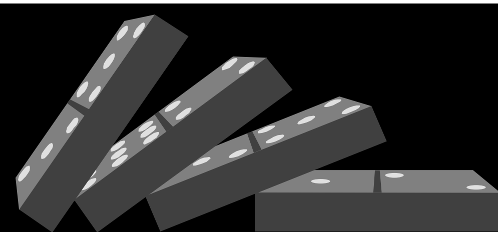
Study Bee

Representation:
Understand the relationship between different representations of the same idea

1
2
3



Study Bee



Induction has many guises

Master their interrelationship

- Formal Arguments
- Loop Invariants
- Recursion
- Algorithm Design
- Recurrences

Study Bee

Exemplification:

Try out a problem or solution on small examples

Study Bee

Abstraction

Abstract away the inessential features of a problem or solution