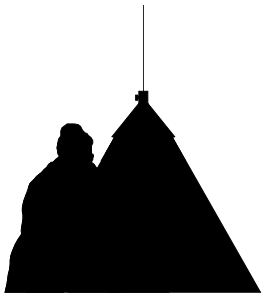
Great Theoretical Ideas In Computer Science

Steven Rudich CS 15-251 Spring 2005

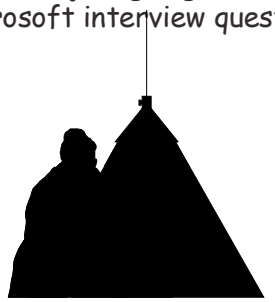
Lecture 15 March 1, 2005 Carnegie Mellon University

Problem Solving: Where does the "aha!" come from?

Relax .....



Relax, I am just going to ask you a Microsoft interview question.



Do You Understand The Question?

Four guys want to cross a bridge that can only hold two people at one time. It is pitch dark and they only have one flashlight, so people must cross either alone or in pairs (bringing the flashlight). Their walking speeds allow them to cross in 1, 2, 5, and 10 minutes, respectively. Is it possible for them to all cross in 17 minutes?

You have one minute to solve this problem

Four guys want to cross a bridge that can only hold two people at one time. It is pitch dark and they only have one flashlight, so people must cross either alone or in pairs (bringing the flashlight). Their walking speeds allow them to cross in 1, 2, 5, and 10 minutes, respectively. Is it possible for them to all cross in 17 minutes?

So what is the answer?

Four guys want to cross a bridge that can only hold two people at one time. It is pitch dark and they only have one flashlight, so people must cross either alone or in pairs (bringing the flashlight). Their walking speeds allow them to cross in 1, 2, 5, and 10 minutes, respectively. Is it possible for them to all cross in 17 minutes?

Intuitive, But False

"10 + 5 + 2 + 1 = 18, so the four guys just can't cross in 17 minutes"

"Even if the fastest guy is the one to shuttle the others back and forth – you use at least 10 + 5 + 2 + 1 > 17 minutes"

Keep track of what you actually know – remember what you merely suspect.

"10 + 5 + 2 + 1 = 18, so it would be weird if the four guys could cross in 17 minutes"

"~~even~~ If we use the fastest guy to shuttle the others, they take too long."

Tagging Strategy

As you talk to yourself, make sure to tag assertions with phrases that denote degrees of conviction

Keep track of what you actually know – remember what you merely suspect.

"10 + 5 + 2 + 1 = 18, so it would be weird if the four guys could cross in 17 minutes"

"~~even~~ If we use the fastest guy to shuttle the others, they take too long."

"~~even~~ If we use the fastest guy to shuttle the others, they take too long."

No faster than 18 solution can use the same "shuttle" guy for every trip. This gives me the idea of trying a solution with 2 shuttle guys.

Any solution must involve more than one guy doing the return trips: it must be that someone gets deposited on one side and comes back for the return trip later!

Intuitively, if we use two shuttles, they should be 1 and 2.

Can I prove that?

Intuitively, if we use two shuttles, they should be 1 and 2.

Can I prove that?
Yes, each shuttle make 3 trips. Thus, he 10 can't be a shuttle. The 5 would use 15 minutes for 2 trip, plus 5 extra for when the 10 goes. 5 can't be a shuttle

1 and 2 must be shuttles.

Let's try a solution with 1 and 2 as the first move….

1 2 5 10
5 10
2 5 10
2
1 2

2 1
1
1 5 10
5 10
1 2 5 10

A different intuitive approach to the same problem:

Load Balancing

5 and 10
Load Balancing:

Handle our hardest work loads in parallel! Work backwards by assuming 5 and 10 walk together.

1 2 5 10
5 10
2 5 10
2
1 2

2 1
1
1 5 10
5 10
1 2 5 10

PROBLEM SOLVING POWER

- UNDERSTAND what the problem is asking.
- TAG thoughts to distinguish between intuition and certainty.
- TRANSFORM/CLEAN-UP intermediate intuitions by pushing them to certainty (theorems).

CERTAINTY AND COMPOSITION

Mathematical statements and inferences are CERTAIN – they compose to make arbitrarily long and completely correct CHAINS of DEDUCTIVE reasoning.

Intuitions are fantastic, but necessarily build on each other inductively to make correct statements. INTUITION are ENABLING, but they are only approximate.

BE OF TWO CLEAR MINDS!

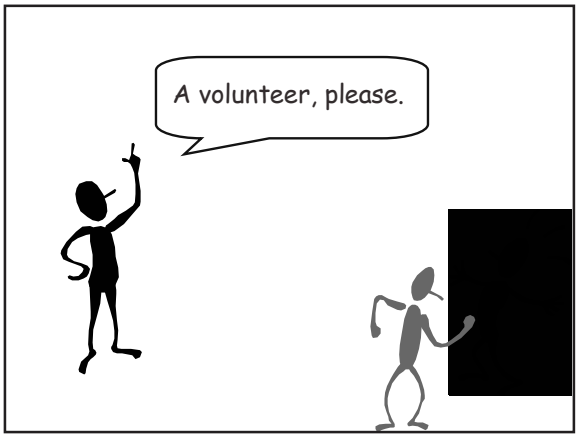
Your intuitive, associative, narrative, conjectural mind.

Your "what do I really know for sure?" – mathematical, logical, and unambiguous mind.

INTUITION and COMPOSITION?

INTUIT
CLEAN-UP
- by PROVING a LEMMA
- By giving the right qualifiers

COMBINE lemmas at will.



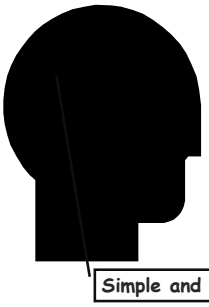
Martial Arts 101

- The novice makes a **huge** motion
- The black belt makes a small motion
- The master makes a tiny motion

Violin, piano, tennis, magic, programming, singing, . . .

- The novice makes a **huge** motion
- The black belt makes a small motion
- The master makes a tiny motion

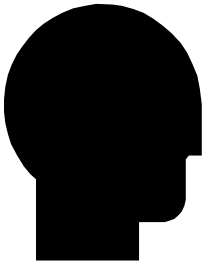
Scanning the brains of master problem solvers



The better the problem solver, the less brain activity is evident. The real masters show almost no brain activity!

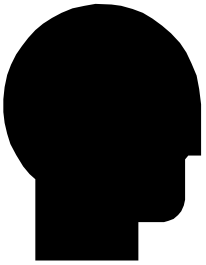
Simple and to the point

Scanning the brains of master problem solvers



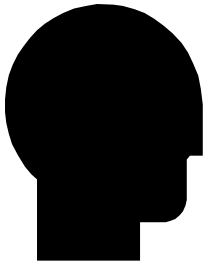
The expert represents his/her state as simply as possible.

Value Simplicity



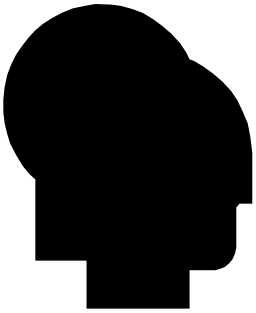
The root of effective thinking, effective science, effective mathematics, effective engineering is to keep the issues simple.

The Dirty Little Secret



Even the smartest people have limited brains that work with simple representations and pictures.

BE OF TWO, SIMPLE MINDS



That really was a Microsoft question.

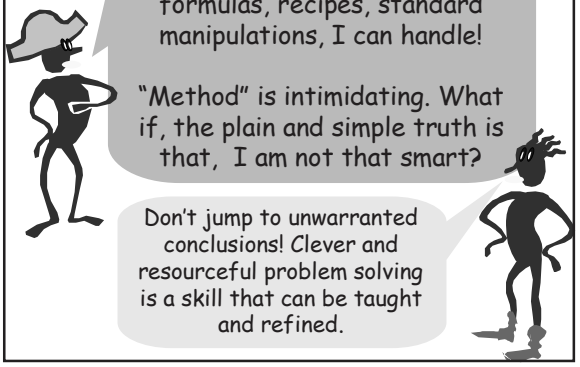
Why do you think that they ask such questions, as opposed to asking for a piece of code to do binary search?

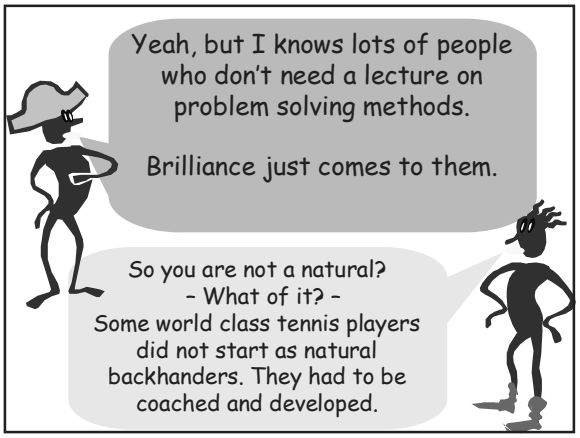
The future belongs to the computer scientist who has

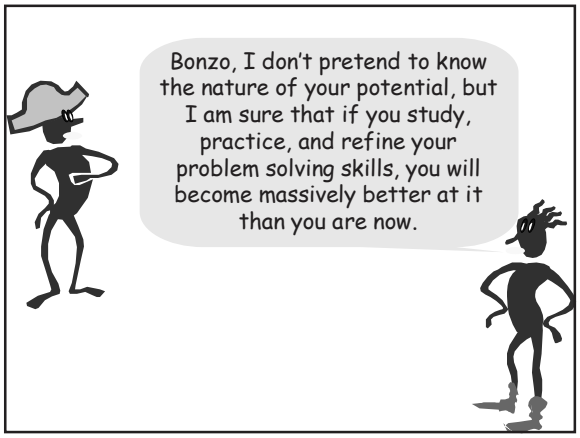
- Content: An up to date grasp of fundamental problems and solutions
- Method: Principles and techniques to solve the vast array of unfamiliar problems that arise in a rapidly changing field

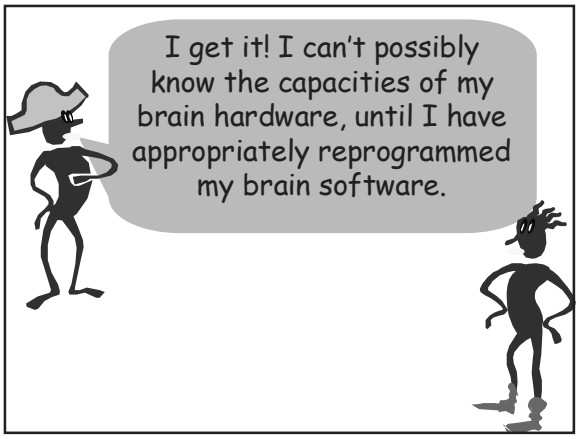
www.discretemath.com

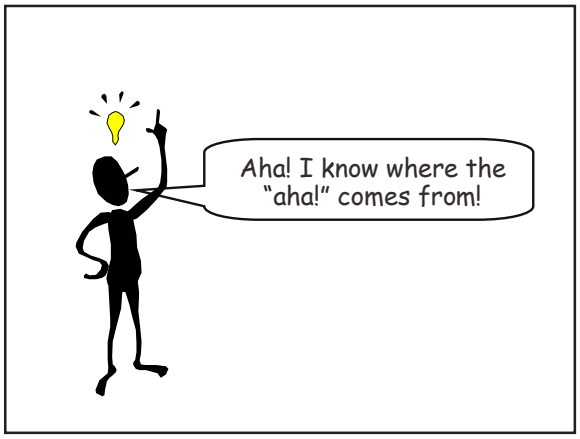
- Content: An up to date grasp of fundamental problems and solutions
- Method: Principles and techniques to solve the vast array of unfamiliar problems that arise in a rapidly changing field

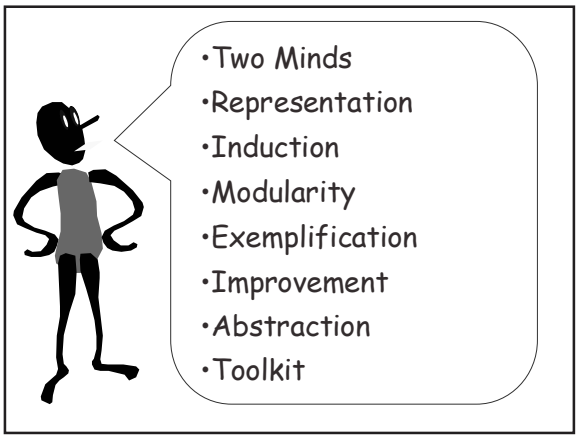












Representation:
Understand the relationship between different representations of the same information or idea

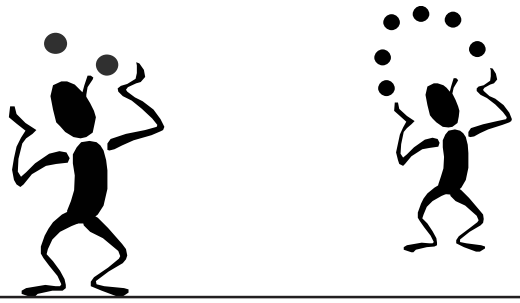
1 ●

2 ● ●

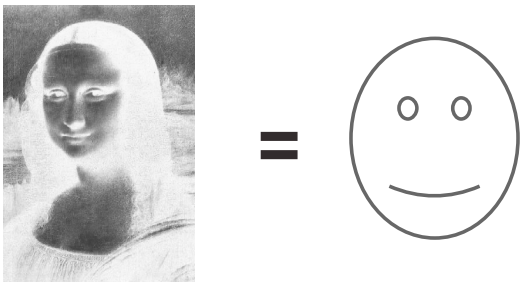
3 ● ● ●

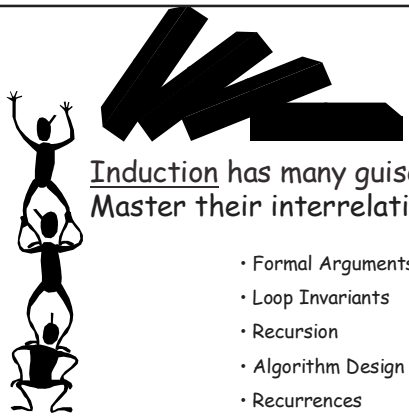
Toolkit:
Name abstract objects and ideas, and put them in your toolkit. Know their advantages and limitations.

Exemplification:
Try out a problem or solution on small examples.



Abstraction:
Abstract away the inessential features of a problem



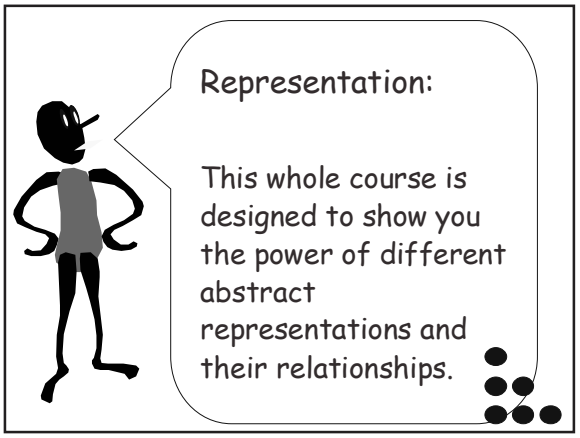


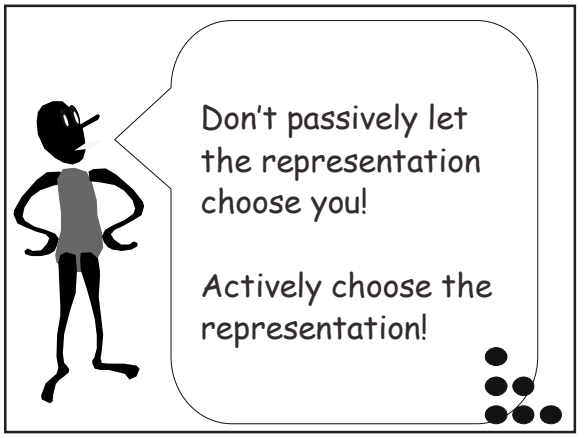
Induction has many guises.
Master their interrelationship.

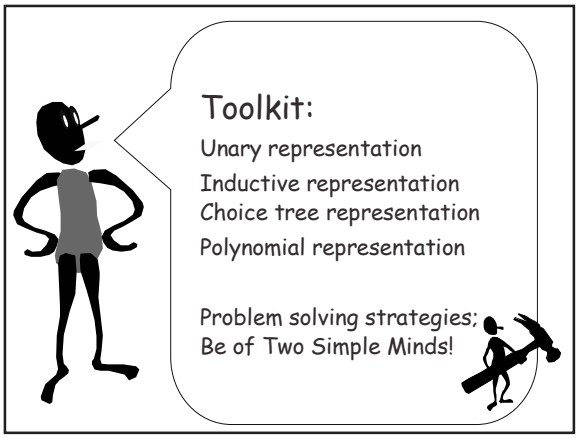
- Formal Arguments
- Loop Invariants
- Recursion
- Algorithm Design
- Recurrences

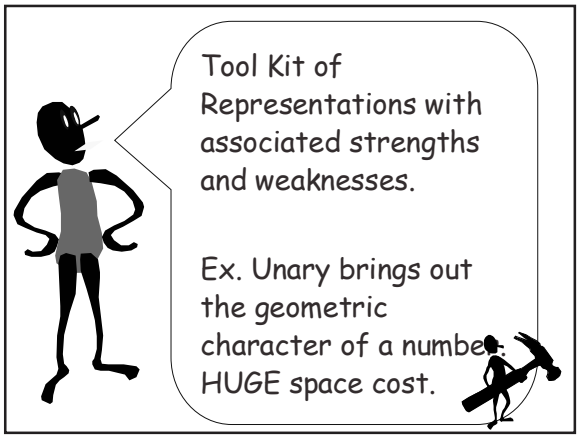
Modularity:
Decompose a complex problem into simpler subproblems

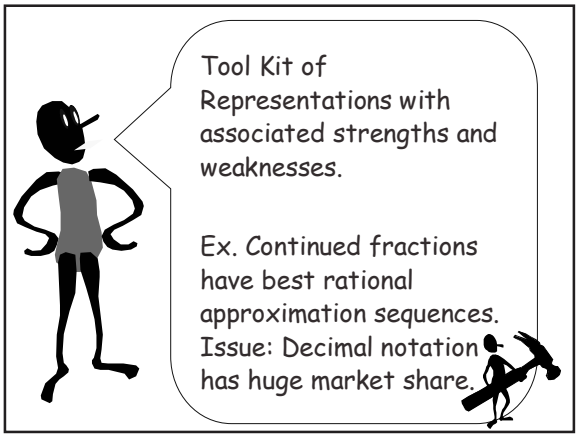
Improvement:
The best solution comes from a process of repeatedly refining and improving solutions and proofs.

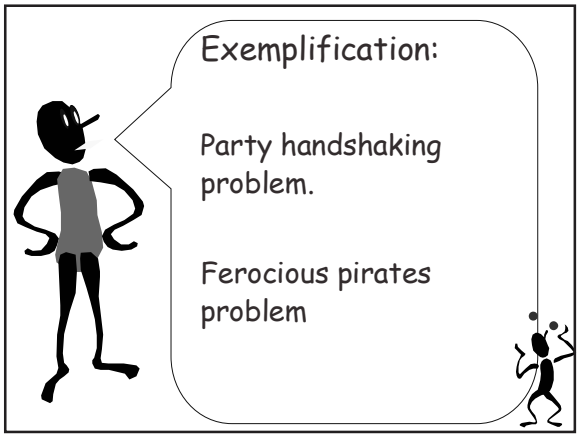

Representation:
This whole course is designed to show you the power of different abstract representations and their relationships.

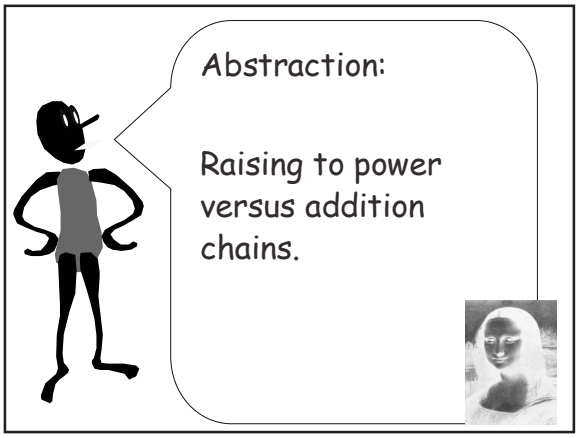

Don't passively let the representation choose you!
Actively choose the representation!

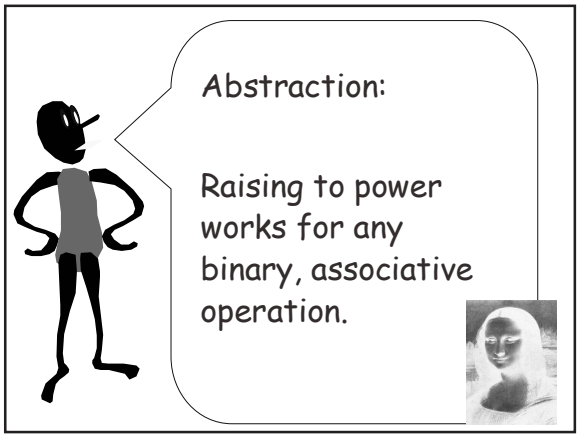

Toolkit:
Unary representation
Inductive representation
Choice tree representation
Polynomial representation
Problem solving strategies;
Be of Two Simple Minds!

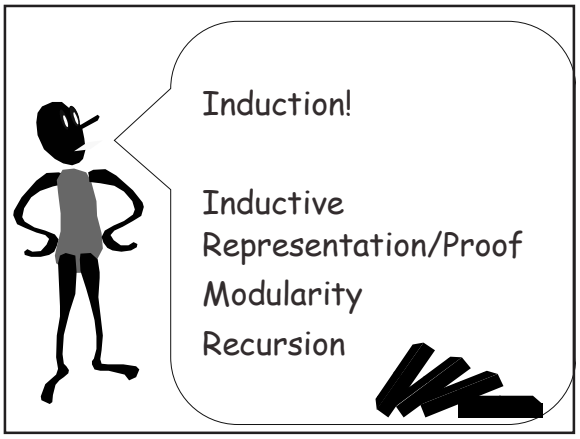

Tool Kit of Representations with associated strengths and weaknesses.
Ex. Unary brings out the geometric character of a number.
HUGE space cost.

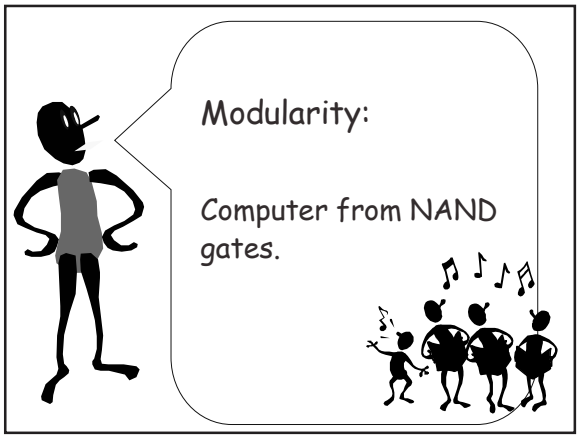

Tool Kit of Representations with associated strengths and weaknesses.
Ex. Continued fractions have best rational approximation sequences.
Issue: Decimal notation has huge market share.

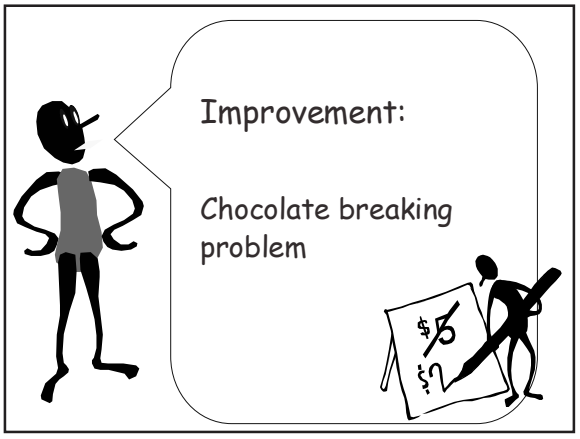

Exemplification:
Party handshaking problem.
Ferocious pirates problem

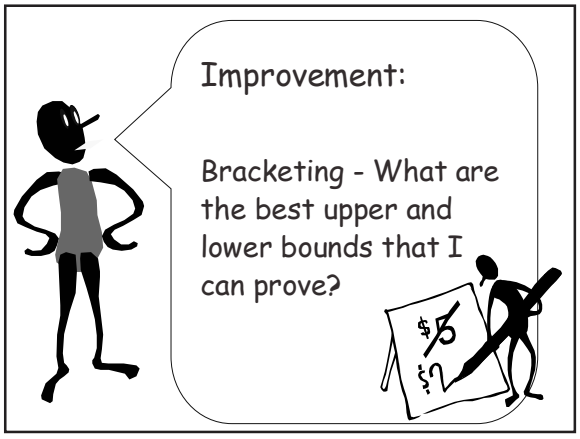

Abstraction:
Raising to power versus addition chains.

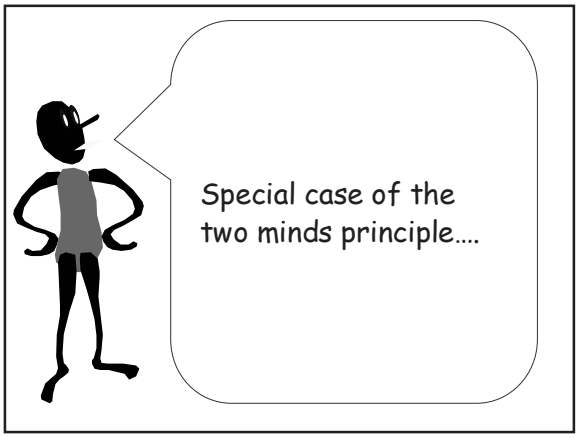

Abstraction:
Raising to power works for any binary, associative operation.


Induction!
Inductive Representation/Proof
Modularity
Recursion

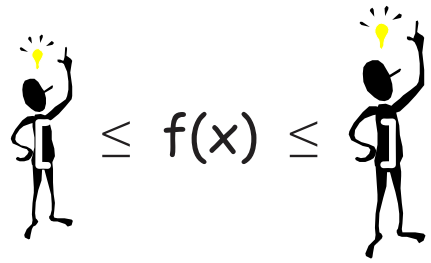

Modularity:
Computer from NAND gates.


Improvement:
Chocolate breaking problem


Improvement:
Bracketing - What are the best upper and lower bounds that I can prove?



Bracketing:
What are the best lower and upper bounds that I can prove?



? o M(30) o 6
? o M(n) o ?

M(n) = numbers of additions required to make n.

? o M(30) o 6
? o M(n) o ?

FIRST CUT: What brackets can you do off the top of your head?

? o M(30) o 6
? o M(n) o ?

SECOND CUT: What brackets can you do in five minutes?

Let $a_0, a_1, a_2, \ldots a_n$, X be input numbers. Goal is to compute :
$a_n X^n + a_{n-1} X^{n-1} + \ldots + a_1 X + a_0$

At any stage we are allowed to ADD or MULTIPLY any two numbers created so far.

How do we MINIMIZE, f(n), the number of operations?

Let $a_0, a_1, a_2, \ldots a_n, X$ be input numbers. Goal is to compute : $a_n X^n + a_{n-1} X^{n-1} + \ldots + a_1 X + a_0$

Intuitively, need one operation to make each power of X, so n-1 < f(n).

We can use n-1 to create the powers of X. Then we can use n to multiply each power by its coefficient. We can use n additions to add them up.
$n-1 \leq f(n) \leq 3n-1$

$P_k(X) = a_n X^k + a_{n-1} X^{k-1} + \ldots + a_{n-k+1} X + a_{n-k}$
Goal: Compute $P_n(X)$

Inductive Goal (Invariant):
Compute $P_0(X), P_1(X), \ldots p_n(X)$

Slick code [Loop invariant $P==P_i(X)$]

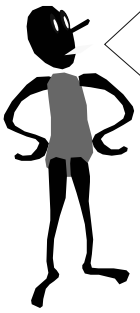
$P := a_n$;
For i = 1 to n do $P := P*X + a_{n-i}$

TOTAL: n mults and n additions. $f(n) \leq 2n$

Let $a_0, a_1, a_2, \ldots a_n, X$ be input numbers. Goal is to compute : $a_n X^n + a_{n-1} X^{n-1} + \ldots + a_1 X + a_0$

At any stage we are allowed to ADD or MULTIPLY any two numbers created so far.

$n-1 \leq f(n) \leq 2n$



AHA! Method:

New problem encountered. Possible representation? Try it out on small examples. Any familiar tools come to mind? Get rid of inessential details. Do some quick bracketing. Inductive viewpoint? Be of two minds!