# Lecture 1: 3/31/2009

## Notes for ECS 20 (Prof. Martel)

**TODAY:**

1. **Introduction**
2. **Some Example Problems**

### 1. Introduction

Prof. Chip Martel, and TA Nick Leaf introduced.
The course homepage is **www.cs.ucdavis.edu/~martel/20**.
A course information sheet is there, and you need to read it.
**Problem Set 1** is will go out on **Thursday**.

**Helpful Background:**
Some math background/sophistication; Basic programming knowledge;

- **Discrete mathematics** deals with finite and countably infinite sets
- Some branches of discrete mathematics are:
  - Combinatorics (how to count things, how to make combinatorial objects that have desired properties)
  - Graph theory (points and two-elements subsets of them)
  - Logic
  - Set theory
  - Probability (again, routinely treated in discrete math classes, but only when we assume that the underlying “probability space” is finite or countably infinite).
  - And much more

**Course Goals**

1. Learn to describe things in precise Math language
2. Define precisely what programs can do.
3. Proof techniques: applications to proving programs do what they are intended to.
4. Program time/space analysis: determine time/space usage.
5. Think recursively.

### 2. Some Example Problems

**Example 1:**
Find the following sum:

$$1 + 2 + \ldots + 100$$

or more generally

$$1 + 2 + \ldots + n$$

**Solution:**

This problem can be viewed algebraically by writing the list of numbers forwards and backwards, i.e.

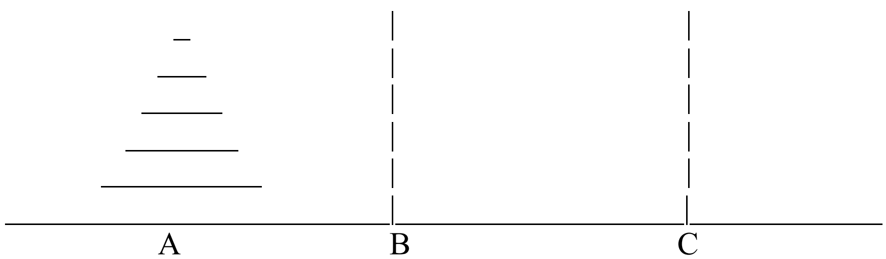
$$
\begin{array}{cccccccccc}
 & 1 & + & 2 & + & \ldots & + & (n-1) & + & n \\
+ & n & + & (n-1) & + & \ldots & + & 2 & + & 1 \\
\hline
 & (n + 1) & + & (n + 1) & + & \ldots & + & (n + 1) & + & (n + 1)
\end{array}
$$

which is $n(n + 1)$

Since this result is really twice the sum we are looking for it follows that

$$1 + 2 + \ldots + n = (n(n + 1)) / 2$$

**Example 2: The Towers of Hanoi**




Five rings of increasing diameter are placed on peg A. The pegs must be moved from A to C using the following rules:

- Only the topmost ring on a peg can be moved.
- A bigger ring cannot be placed on a smaller one.

Problem: Find a function that describes the least number of moves needed to solve the problem when you have n rings.

How many moves do you think it takes to move the five rings. One student knew that the answer is 31.

We want a formula that specifies a number of moves that is both:

- **Sufficient**: there **is** a solution to the game using this number of moves.
- **Necessary**: no solution can use **fewer** moves than this.

**Solution:**

First, let’s define what we’re interested in. Let

$T_n$ = The minimum number of moves needed to move the n rings from peg A to peg C (obeying the rules of the game).

Actually, this isn’t good. We need to generalize the definition to do the job. So, instead, let

$T_n$ = The minimum number of moves needed to move n rings from some one specified peg to some other specified peg (obeying the rules of the game).

**Sufficiency**:

Think recursively. Assume a “black box” algorithm can move the first n - 1 rings from any peg to any other peg. Solving the problem this way requires that the first n -1 rings be moved, then the largest ring be moved once, then the smaller rings be moved on to the largest ring. This number of moves can be represented by:

$$T_n \leq T_{n-1} + 1 + T_{n-1} = 2T_{n-1} + 1$$

**Necessity**:

Now we have to reason about *any* algorithm that solves the puzzle.
Any solution must move the largest ring to the final peg for the very last time. That takes one move. But before that happened, we had to get the n-1 rings that were formerly on top of the start peg and move them off to a free peg. That takes at least $T_{n-1}$ moves. After we got the biggest ring to its destination peg, we had to move the n-1 smaller rings from the free peg where they were at to the final peg. That takes at least $T_{n-1}$ moves. So, all in all, any solution needs to spend at least

$$T_n \geq 1 + T_{n-1} + T_{n-1} = 2T_{n-1} + 1$$

moves.

Putting together the two inequalities we have that

$$T_n = 2T_{n-1} + 1$$

We know that in order to move zero rings to their final location requires zero moves, so

$T_0 = 0$

Using this as our base value we can then determine that:

| $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | … | $T_n$ |
|---|---|---|---|---|---|---|
| 1 | 3 | 7 | 15 | 31 | | $2^n - 1$ (apparently) |

One way to get the general formula is by repeated substitution:

$$T_n = 2\,T_{n-1} + 1$$
$$= 2\,[2\,T_{n-2} + 1] + 1 = 2^2\,T_{n-2} + (1 + 2)$$
$$= 2^2\,[2\,T_{n-3} + 1] + (1 + 2) = 2^3\,T_{n-3} + (1 + 2 + 4)$$

$$= 2^n * 0 + (1 + 2 + 2^2 + \ldots + 2^{n-1})$$
$$= (1 + 2 + 2^2 + \ldots + 2^{n-1}) = 2^n - 1$$

Binary Representation

**Example 2:**

Prove that $x = \sqrt{2}$ is **irrational**. (see page 2 of the text for definitions of R, Z, Q and N).

Definition:

x Є **R** is **rational (in the set Q)** if x = p/q for some integers p and q Є **Z**, $q \neq 0$.

x Є **R** is **irrational** if it is not rational.

Note: "if" in definitions mean "if and only if", or "exactly when"

Here we prove by contradiction.
Assume for contradiction that x *is* rational, ie,

x = p/q for some integers p and q, $q \neq 0$

Without a loss of generality, it can be assumed that either p is odd or q is odd
(if they're both even, cross out the common factors of 2 until one of the numbers is odd).
Additionally, an odd number squared is still an odd number.

$\sqrt{2} = p/q$ square both sides:
$\rightarrow\ 2 = p^2/q^2$ multiply through by the denominator:
$\rightarrow\ 2q^2 = p^2$ (*)

From this we know that p is even, because the square of an odd number is odd (why?!). So we can write

p = 2j for some j Є **Z** and substituting 2j for p in (*) above we get:

$$2q^2 = (2j)^2$$

$$\rightarrow 2q^2 = 4j^2$$

$$\rightarrow q^2 = 2j^2$$

so by the same reasoning as before, q must be odd.

The contradiction is that we have now shown both p and q are even, but we assumed at least one was odd. We can conclude that our original assumption is wrong: $x = \sqrt{2}$ is **not** equal to any p/q, so is not rational, which, by definition, means that it is **irrational**.

**Example 3:**

We claim that : 5 shuffles of a deck of cards is **not** enough to randomize the cards.

Here by **shuffles** I mean the usual "rifle shuffle."

For example for a simple deck of 8 cards 1 2 3 4 5 6 7 8 the shuffle breaks it into two groups (the top and bottom) e.g. 1 2 3 4 and 5 6 7 8 (though they need not be equal) and interleaves the two groups: e.g. 1 5 6 3 7 4 8 to get a new order.

Even though we won't define what it means to randomize the cards, clearly a deck cannot be well randomized unless you can get **any** resulting sequence of cards, including, for example, the sequence:

52, 51, …, 3, 2, 1

A stronger requirement (we won't use) is that each of the 52! different orders is equally likely to be produced.

We are going to show that 5 shuffles of this deck will **never** transform the specified starting sequence to the specified final sequence. So it can't do a good job of mixing the deck. More next class.