CS 241
Data Organization using C

How to Solve it

Instructor: Joel Castellanos
e-mail: joel@unm.edu
Web: http://cs.unm.edu/~joel/
Office: Farris Engineering Center 319

4/9/2015

How to Solve it

"If there's a problem you can't solve, then there is a simpler problem that you can solve. Find it!"

-- George Pólya (1887 –1985)
Professor of Mathematics at Stanford University
Scientific Thinking

- Work through examples using pencil and paper: see what works and what doesn't.

- When you've seen enough examples, formulate a general algorithm.

- Try to think of counterexamples to that algorithm.

Hard Problems

In order to learn scientific thinking, you need hard problems.

Problems that are too hard to solve all at once.

Whenever you are working near your limits, sometimes you will face a problem that, for now, is beyond those limits.
The Computer Science Club

If you find these problems hard, do not think you are dumb or that you are not "cut out" to be a computer scientist.

The problems are designed to be hard.

The person sitting next to you may have been programming since grade school, or have years of professional experience.

Learning Problem Solving

Solving problems is a practical skill like, for example, swimming.

We acquire any practical skill by observing, imitating and practicing.

Trying to swim, you imitate what other people do with their hands and feet to keep their heads above water.

At first, you splash around a lot.

Finally, you become a good swimmer by practicing swimming.
The Simple Way

- Find first the *easiest* way, not the *best* nor most *efficient* way.
  - Easiest path to the solution needed now.
  - Easiest for you to understand 3 years from now.
  - Easiest for someone else to understand.
  - Easiest to debug.
  - Easiest to expand.
  - Easiest to reuse on a different problem.

- Usually, a Programmer’s labor-hours cost more than a computer’s milliseconds.

- After you get the ideas working together, if not satisfied with run speed, *then* think about optimization.

Solve it with Modules

- Each module can be written independently.
- Each module can be debugged independently.
- Each module can be expanded independently.
- *If optimization is needed*, each module can be optimized independently.
"The reason I'm successful is because I'm lucky. But I didn't get lucky until I started working 90 hours a week."

"My dad says persistence is the key to success. So I'm going to keep giving you the same wrong answer until it becomes the right answer!"
It is tempting to imagine solving a problem with advanced tools you do not yet understand, and to imagine the solution to be glorious.

This is chancing the wind.

Try first to actually solve the problem with the tools you do know - even if you believe it will not be the best of all possible solutions.

If you cannot, then learn one new thing and use it to solve a smaller problem.