

Doing Research in Software Analysis

Lessons and Tips

Zhendong Su
UC Davis

Tip #0

fortune cookie @ last night's dinner

**Your success in life must be
earned with earnest efforts.**

A bit of my background



- Math BA & CS BS @ UT Austin
- CS PhD in 2002 @ Berkeley
 - ▣ Constraint and type based program analysis
 - ▣ Early: practical applications
 - ▣ Mid: constraint solving
 - ▣ Final: pure algorithms & complexity
- CS Professor @ UC Davis (since 2003)
 - ▣ Many interests, but stay in PL/SE/Security mostly
 - ▣ Whatever excites me AND my students

Lesson #1

Get inspired!



**Dijkstra's Three Golden Rules
for Successful Scientific Research
(The Idealist's View)**

Dijkstra's Golden Rule #1



Raise your quality standards as high as you can live with, avoid wasting your time on routine problems, and always try to work as closely as possible at the boundary of your abilities. Do this, because it is the only way of discovering how that boundary should be moved forward.

Dijkstra's Golden Rule #2



We all like our work to be socially relevant and scientifically sound. If we can find a topic satisfying both desires, we are lucky; if the two targets are in conflict with each other, let the requirement of scientific soundness prevail.

Dijkstra's Golden Rule #3



Never tackle a problem of which you can be pretty sure that (now or in the near future) it will be tackled by others who are, in relation to that problem, at least as competent and well-equipped as you.



Read & Reread
Richard Hamming
“You and Your Research”

What is great research?

- Two styles of **great research**
 - ▣ A longstanding **hard nut**, you **cracked** it
 - ▣ **Seminal** work that **opens** up a **new area**
- Great research should have
 - ▣ Given a new powerful **solution** / **concept**
 - ▣ The **potential** to change how
 - A community **thinks** about a particular problem
 - People **approach** things

Lesson #2

Pick a problem *for you*

Pick a good problem



- Very **hard**
 - Art
 - Taste
 - Luck
 - Pain killers vs. vitamins

- Bottom-line
 - **You should care about its solution!**
 - **You are just excited about it!**

Types of problems: established



- The easiest kind
- Examples: pointer analysis, race detection ...
- **Incremental**, but **solid bricks**
- **Most** good work falls here
- **Most** bad work falls here as well
 - Incremental, but not quite solid bricks

Types of problems: hard nuts



- **Longstanding**, well-known problems
- New techniques
- **Breakthroughs**
- The **problem solver** type

Types of problems: essence



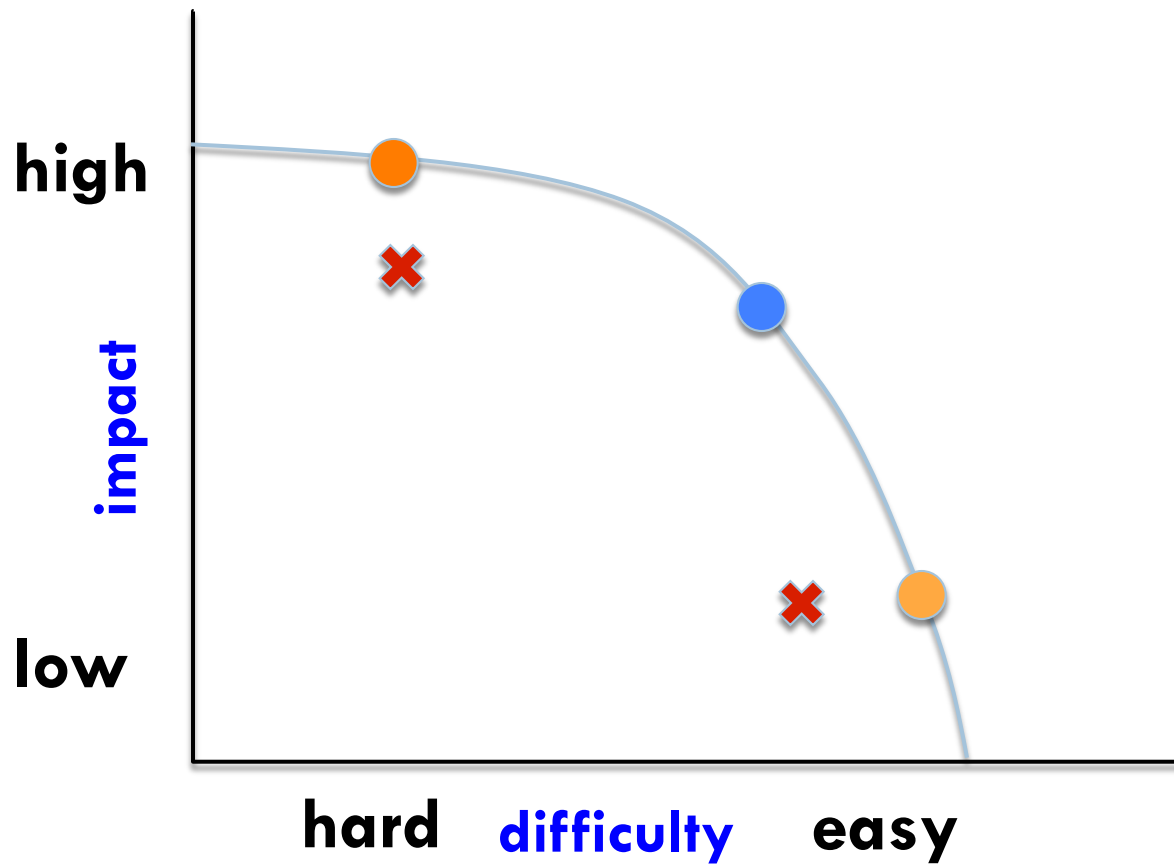
- Take a real-world **messy** problem
- Understand its **essence**
- Formulate/conceptualize
 - Problem
 - Concept
 - Solution

Types of problems: the hardest



- Problems **people don't even know** they have
- The **visionary**, abstract **thinker** type
- **Revolutions** typically come out of here

The Pareto front



Lesson #3

Understand, justify, execute

How to solve a problem (1)

- **Understand** it
 - A *well-chosen example* for motivation & illustration
 - Never underestimate its importance
- **Formalize** if possible
 - Goal: obtain a **crisp description** of the problem
- What are the **key difficulties**
 - Conceptual, technical, and engineering
 - Extremely important to understand these well

How to solve a problem (2)

- How to approach the challenges
 - ▣ **Key insights**
 - ▣ **Key technical novelties**
 - ▣ **Key engineering hurdles** (i.e., feasibility)
- How to evaluate if you succeed doing the work
 - ▣ Ask this question early, don't delay
- What **unique** & **impressive** to show at the end
 - ▣ Vision, problem formulation, theory, technique, results?
 - ▣ Is it so evidently unique and impressive?
 - ▣ Any elements of intrigue & surprise

How to solve a problem (3)

- The process is **full of decisions**
 - **Understand** your **options** well
 - **Don't rush** to take the first option you think of
 - Can **justify** your **choices**
 - Also **understand** what's **important**, what's not
 - **Avoid** taking **shortcuts**
- Takeaways
 - Always **mindful**
 - Justify, **justify**, and justify
 - **Convince** yourself first, then others

Tip #1

Easy vs. difficult

Tip #2

Hammer vs. nail

Tip #3

Release tools

Tip #4

Excitement vs. despair

Tip #5

Results vs. the process

Final Remarks

- Everybody can do **fun & great** work
 - When you **believe** so
 - Why not, we all should!
 - When you **put your heart** to it

- **Great things will happen,** just
 - **Dream** big, **think** big
 - Be a **mindful** & **flexible thinker**
 - **Work hard, never give up**