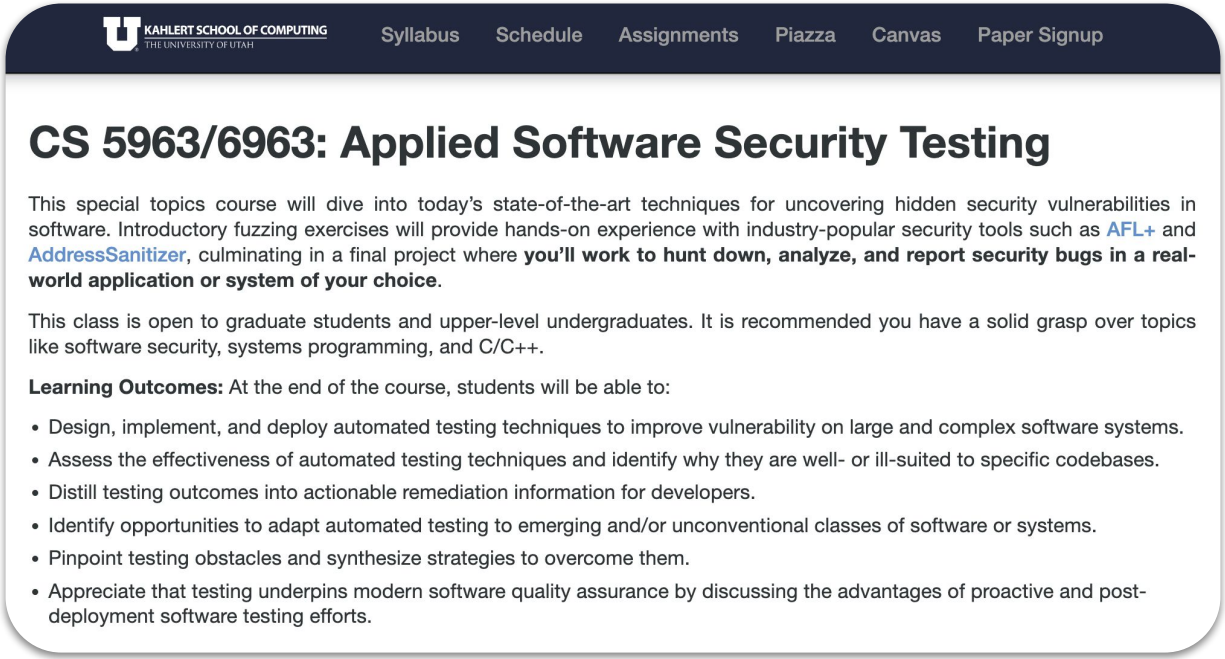
Week 1: Lecture B

Research 101: Ideas

Wednesday, January 10, 2024

Recap: Course Website

cs.utah.edu/~snagy/courses/cs5963



Recap: Course Resources

Course website ..... assignments, schedule, slides, paper signup

Piazza ..... questions, discussion, announcements

Canvas ..... homework submission, course gradebook

Instructor email (snagy@cs.utah.edu**)** ..... administrative issues

Recap: Lateness Policy

- Assignments will be posted on course website
  - See cs.utah.edu/~snagy/courses/cs5963/assignments
- Due by **11:59 PM** on the specified deadline date
  - Late assignments will **not** be accepted
- If you are sick / traveling / abducted by aliens…
  - Try to keep me posted and we will figure something out

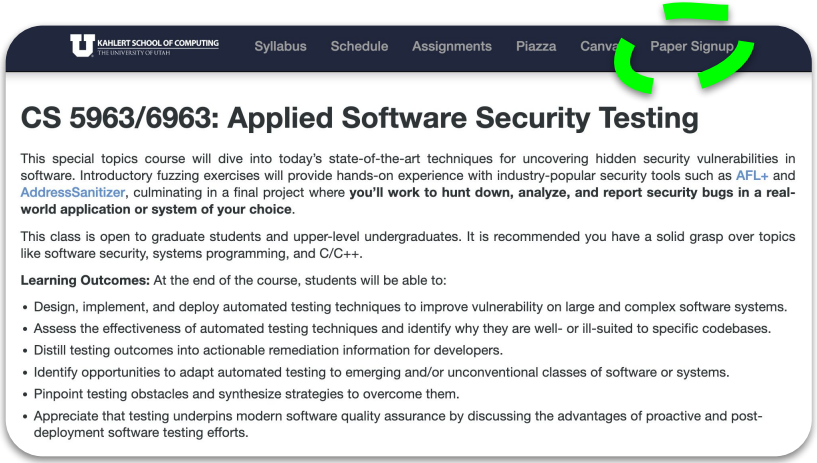
Recap: Course Materials

- No textbook is required for this course
- Some excellent resources on fuzzing are:
  - **The Fuzzing Book** by Zeller, Gopinath, Böhme, Fraser, and Holler
  - **Fuzzing Against the Machine** by Antonio Nappa and Blazquez
- Other general computer security textbooks:
  - **Introduction to Computer Security** by Goodrich and Tamassia
  - **Security Engineering** by Ross Anderson
- These are are linked on the course syllabus
  - cs.utah.edu/~snagy/courses/cs5963/

Recap: No Exams

Recap: Paper Presentations

- **Signup sheet** available on course website (must use **UofU gcloud** account)
  - **38 fuzzing papers** from top venues in security, software engineering, and some workshops
  - Choose one paper by **Monday, January 22**



KAHLERT SCHOOL OF COMPUTING
THE UNIVERSITY OF UTAH

Syllabus | Schedule | Assignments | Piazza | Canvas | Paper Signup

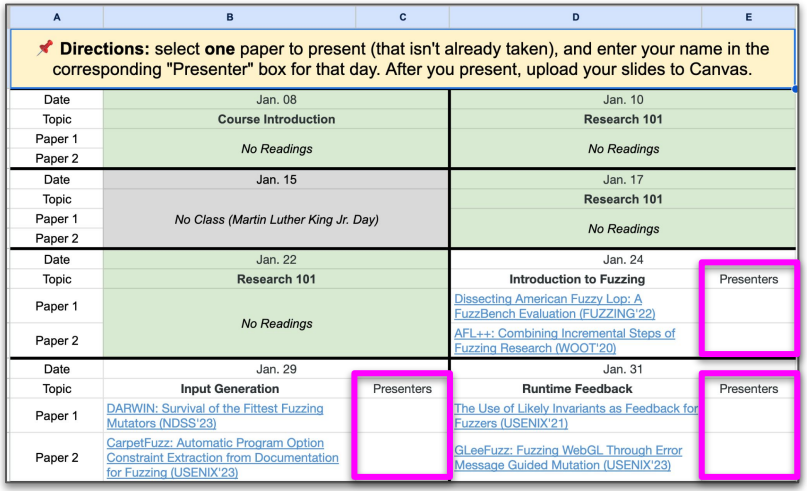
CS 5963/6963: Applied Software Security Testing

This special topics course will dive into today's state-of-the-art techniques for uncovering hidden security vulnerabilities in software. Introductory fuzzing exercises will provide hands-on experience with industry-popular security tools such as AFL+ and AddressSanitizer, culminating in a final project where **you'll work to hunt down, analyze, and report security bugs in a real-world application or system of your choice**.

This class is open to graduate students and upper-level undergraduates. It is recommended you have a solid grasp over topics like software security, systems programming, and C/C++.

Learning Outcomes: At the end of the course, students will be able to:

- Design, implement, and deploy automated testing techniques to improve vulnerability on large and complex software systems.
- Assess the effectiveness of automated testing techniques and identify why they are well- or ill-suited to specific codebases.
- Distill testing outcomes into actionable remediation information for developers.
- Identify opportunities to adapt automated testing to emerging and/or unconventional classes of software or systems.
- Pinpoint testing obstacles and synthesize strategies to overcome them.
- Appreciate that testing underpins modern software quality assurance by discussing the advantages of proactive and post-deployment software testing efforts.



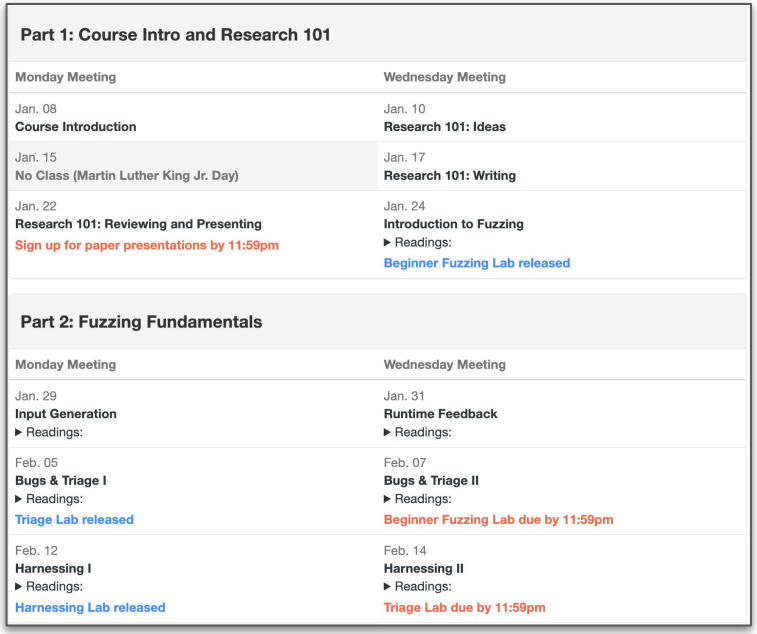
📌 **Directions:** select **one** paper to present (that isn't already taken), and enter your name in the corresponding "Presenter" box for that day. After you present, upload your slides to Canvas.

	A	B	C	D	E
1	📌 **Directions:** select **one** paper to present (that isn't already taken), and enter your name in the corresponding "Presenter" box for that day. After you present, upload your slides to Canvas.				
2	Date	Jan. 08		Jan. 10	
3	Topic	**Course Introduction**		**Research 101**	
4	Paper 1	*No Readings*		*No Readings*	
5	Paper 2				
6	Date	Jan. 15		Jan. 17	
7	Topic	*No Class (Martin Luther King Jr. Day)*		**Research 101**	
8	Paper 1			*No Readings*	
9	Paper 2				
10	Date	Jan. 22		Jan. 24	
11	Topic	**Research 101**		**Introduction to Fuzzing**	Presenters
12	Paper 1	*No Readings*		Dissecting American Fuzzy Lop: A FuzzBench Evaluation (FUZZING'22)	
13	Paper 2			AFL++: Combining Incremental Steps of Fuzzing Research (WOOT'20)	
14	Date	Jan. 29		Jan. 31	
15	Topic	**Input Generation**	Presenters	**Runtime Feedback**	Presenters
16	Paper 1	DARWIN: Survival of the Fittest Fuzzing Mutators (NDSS'23)		The Use of Likely Invariants as Feedback for Fuzzers (USENIX'21)	
17	Paper 2	CarpetFuzz: Automatic Program Option Constraint Extraction from Documentation for Fuzzing (USENIX'23)		GLeeFuzz: Fuzzing WebGL Through Error Message Guided Mutation (USENIX'23)	

Recap: Key Dates

- **Jan. 15** No class (MLK Jr. Day)
- **Jan. 22** **Select one paper to present**
- **Feb. 07** Lab 1 due
- **Feb. 14** Lab 2 due
- **Feb. 19** No class (President's Day)
- **Feb. 28** Lab 3 due
- **Feb. 28** **5-minute project proposals**
- **Mar. 04 & 06** No class (Spring Break)
- **Apr. 17 & 22** **Final project presentations**

cs.utah.edu/~snagy/courses/cs5963/schedule



Part 1: Course Intro and Research 101

Monday Meeting	Wednesday Meeting
Jan. 08 **Course Introduction**	Jan. 10 **Research 101: Ideas**
Jan. 15 **No Class (Martin Luther King Jr. Day)**	Jan. 17 **Research 101: Writing**
Jan. 22 **Research 101: Reviewing and Presenting** Sign up for paper presentations by 11:59pm	Jan. 24 **Introduction to Fuzzing** ▸ Readings: Beginner Fuzzing Lab released

Part 2: Fuzzing Fundamentals

Monday Meeting	Wednesday Meeting
Jan. 29 **Input Generation** ▸ Readings:	Jan. 31 **Runtime Feedback** ▸ Readings:
Feb. 05 **Bugs & Triage I** ▸ Readings: Triage Lab released	Feb. 07 **Bugs & Triage II** ▸ Readings: Beginner Fuzzing Lab due by 11:59pm
Feb. 12 **Harnessing I** ▸ Readings: Harnessing Lab released	Feb. 14 **Harnessing II** ▸ Readings: Triage Lab due by 11:59pm

Questions?

This time on CS 5963...

Research 101: Ideas

What is “Research”?

This →

Generating an Idea

Pursuing an Idea

Also this ←

Course Goals

- Help you become **better researchers**
- Expose you to **different perspectives**
- Experience with **state-of-the-art tools**
- Get course credit so you can graduate?
- All while learning about **software testing**

Course Goals

- Help you become **better researchers**
- Expose you to **different perspectives**
- Experience with **state-of-the-art tools**
- Get course credit so you can graduate?
- All while learning about **software testing**

Reading / evaluating
published research

Conducting / presenting
your own research

Research 101:
Ideas, Writing, Reviewing, and Presenting Research

Ideas:
The Foundation of Research

What are “Ideas”?

Ideas underpin research...

- Great ideas can be **ruined** **by bad execution**
  - Think of every neat Shark Tank product... with poor sales

Ideas underpin research...

- Great ideas can be **ruined** **by bad execution**
  - Think of every neat Shark Tank product... with poor sales
- Great execution **cannot promote a poor idea**
  - Think of every dumb Shark Tank product... with good sales

Ideas underpin research…

- Great ideas can be **ruined** **by bad execution**
  - Think of every neat Shark Tank product… with poor sales
- Great execution **cannot promote a poor idea**
  - Think of every dumb Shark Tank product… with good sales
- Great ideas are **context and time sensitive**
  - There is a “right time” for specific ideas
  - If you move too slow, you’ll get scooped!

Great ideas underpin **great research**

Where do great ideas come from?

How to come up with research ideas?

Asked 9 years, 8 months ago Modified 5 years, 3 months ago Viewed 85k times

120

128

As a very new researcher who is exploring the best way to generate ideas, some guidance on this question would be very helpful. I have found that this is NOT easy. Ideas seem to pop out of my Professor every day and I wonder how he does it. This question is broad;

- How do you tend to come up with initial/seed ideas? What is your search method (if you have one)?
- What proportion of your ide ome from; (i) colleagues, (ii) intentionally browsing the literature for inspiration, (iv) conferences, (v) other?
- How do you prioritize rese
- Is there any special, genera discovered to sift out those ideas that are likely to be unrealistic e a generation?

https://academia.stackexchange.com/questions/5853/how-to-come-up-with-research-ideas

Where do great ideas come from?

How to com

Asked 9 years, 8

120

128

As a ve uidance on this que to pop out of my Prof

- Ho nod (if you hav
- Wh ntentionally bro (v) other?
- Ho
- Is t nose ideas that are

https://academia.stackexchange.com/questions/5853/how-to-come-up-with-research-ideas

Generating Ideas

Problems are *not* ideas...

… but ideas *emerge* from problems!

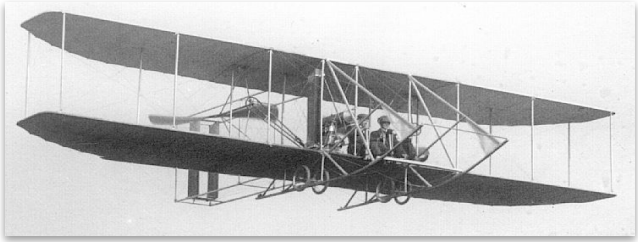
- What is the **next big problem** that society will face?
- What are the **unsolved challenges** in a specific area?
- Are there any common yet unsupported **assumptions**?
- Does existing approaches have **unnecessary hurdles**?

Research Directions

Research Projects

Big vs. Small Ideas

- **Big ideas:** research directions or arcs
  - What grant proposals are centered around
  - Usually generated solo or with 1–2 others
  - Cannot be too complex—must be realistic
  - Often leads to—or requires—many small ideas



Big vs. Small Ideas

- **Big ideas:** research directions or arcs
  - What grant proposals are centered around
  - Usually generated solo or with 1–2 others
  - Cannot be too complex—must be realistic
  - Often leads to—or requires—many small ideas

- **Small ideas**: one small step forward
  - Concrete, “paper-sized” research projects
  - Usually generated with your collaborators
  - Often the projects assigned to grad students
  - Spin-offs of—or the inspiration behind—big ideas

Themes of Ideas

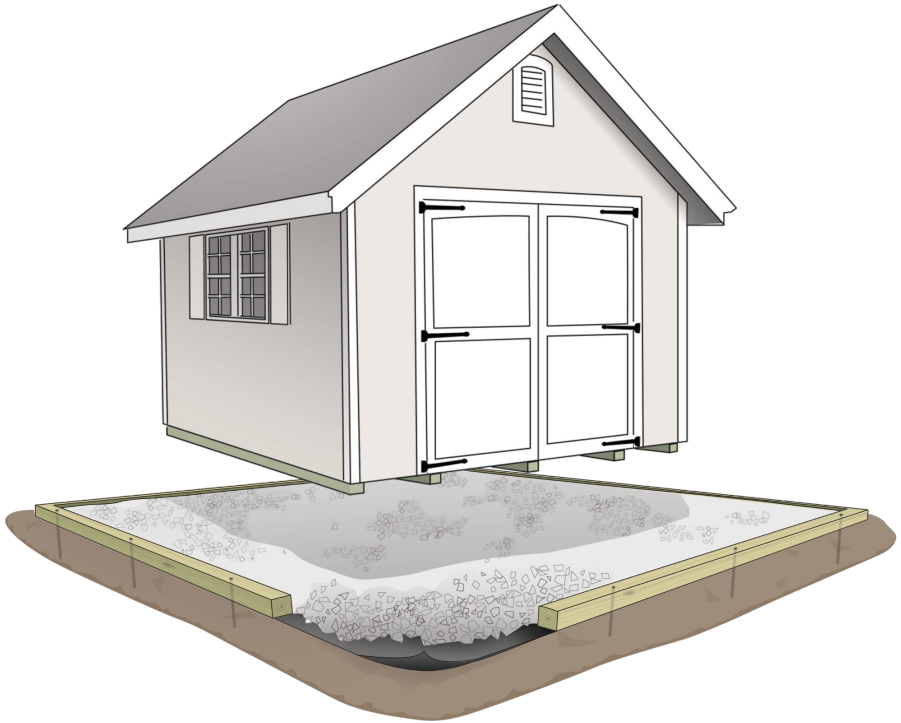
Observation: That's odd... ?

Curiosity: What happens if... ?

Challenge: How do I do... ?

Transference: That works there... will it work here?

Before you brainstorm...

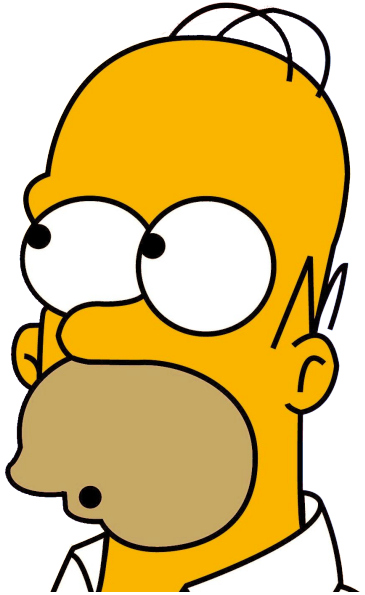


The foundation of good ideas is **understanding**

- How something works
- What assumptions it makes
- How your approach would work
- Why a problem is worth solving

Seeking Inspiration

- **Read technical papers**
- Read blogs and news sites
- Attend technical presentations
- **Talk with your colleagues**
- Volunteer to review papers
- Present already-published work
- Work with those in industry
- **Do something other than research**



Seeking Inspiration

- **Categorize previous work:** look for patterns or missed opportunities

Execution Mechanism	Fuzzing Implementations	Level of Efficiency		Execution Correctness	Windows Kernel Compatibility
		Target	Kernel		
Process Creation	WinAFL [18], Manul [35], KillerBeez [36],	✗	✗	✔	full
Forkserver-based Cloning	Winnie [19]	✔	✗	✔	partial
In-memory Looping	WinAFL [18], TinyAFL [37], Jackalope [38]	✔	✔	✗	full
Kernel-based Snapshotting	AFL++ LKM [17], Xu et al. [13], Zhao et al. [16]	✔	✔	✔	none

Fuzzer	AFL	WinAFL	HonggFuzz	Peach
Feedback	✔	✔	✗	✗
Forkserver	✔	✗	✗	✗
Open-source	✔	✔	✔	✗
Windows	✗	✔	✔	✔

	Technique	Fast	SB	Ptr Arith.			Lack of metadata				Instrum.		Archs
				C1	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*	*C8*	*C9*	*C10*	
RetroWrite [19]	symbolization	✓	✓	✓							✓	✓	✓	x64
Repica [23]	symbolization	✓	✓	✓				✓			✓	✓	✓	aarch64
Egalito [47]	IR lifting	✓	✓	✓				✓			✓	✓	✓	aarch64, x64
DDisasm [22]	symbolization	✓	✓	✓			✓	✓		*[1]	✓	✓		aarch64, x64
ICFGP [32]	trampolines	✓	✓	✓		✓	✓	✓		✓	✓	✓		aarch64, x64, ppc
StochFuzz [50]	stoch. rewriting	N/A[3]		✓	✓	✓	✓	✓		✓	✓	✓	✓	x64
E9Patch [20]	trampolines		✓	✓	✓	✓	✓	✓		✓	✓		✓	x64
Multiverse [12]	recompilation		✓	✓	✓	✓	✓	✓	✓	*[1]	*[2]	✓	✓	x64

Seeking Inspiration

- **Categorize previous work:** look for patterns or missed opportunities

Execution Mechanism	Fuzzing Implementations	Level of Efficiency: Target	Level of Efficiency: Kernel	Execution Correctness	Windows Kernel Compatibility
Process Creation	WinAFL [18], Manul [35], KillerBeez [36],	✗	✗	✔	full
Forkserver-based Cloning	Winnie [19]	✔	✗	✔	partial
In-memory Looping	WinAFL [18], TinyAFL [37], Jackalope [38]	✔	✔	✗	full
Kernel-based Snapshotting	AFL++ LKM [17], Xu et al. [13], Zhao et al. [16]	✔	✔	✔	none
Target-embedded Snapshotting	WinFuzz	✔	✔	✔	full

Fuzzer	AFL	WinAFL	HonggFuzz	Peach	WINNIE
Feedback	✔	✔	✗	✗	✔
Forkserver	✔	✗	✗	✗	✔
Open-source	✔	✔	✔	✗	✔
Windows	✗	✔	✔	✔	✔

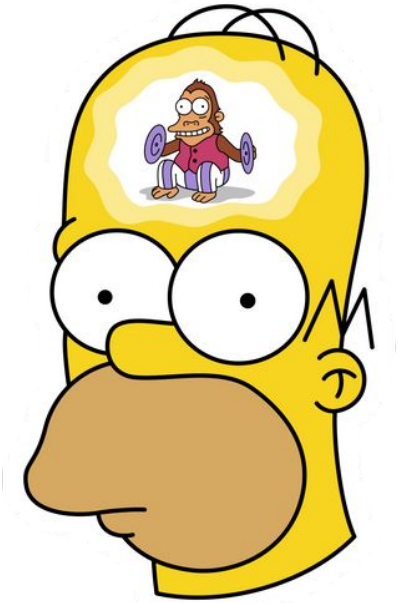
	Technique	Fast	SB	Ptr Arith. C1	C2	C3	Lack of metadata C4	C5	C6	C7	C8	Instrum. C9	C10	Archs
RetroWrite [19]	symbolization	✓	✓	✓							✓	✓	✓	x64
Repica [23]	symbolization	✓	✓	✓				✓			✓	✓	✓	aarch64
Egalito [47]	IR lifting	✓	✓	✓				✓			✓	✓	✓	aarch64, x64
DDisasm [22]	symbolization	✓	✓	✓			✓	✓		*[1]	✓	✓		aarch64, x64
ICFGP [32]	trampolines	✓	✓	✓		✓	✓	✓		✓	✓	✓		aarch64, x64, ppc
StochFuzz [50]	stoch. rewriting	N/A[3]		✓	✓	✓	✓	✓		✓	✓	✓	✓	x64
E9Patch [20]	trampolines		✓	✓	✓	✓	✓	✓		✓	✓		✓	x64
Multiverse [12]	recompilation		✓	✓	✓	✓	✓	✓	✓	*[1]	*[2]	✓	✓	x64
ARMore	symbolization	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	*[4]	aarch64


Finding great
ideas is like
mining gold

Cultivating Ideas

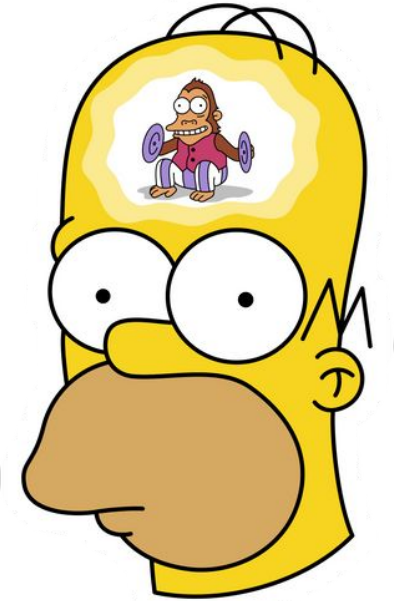
You won’t think of an idea twice!

- **You will forget ideas**
  - You will also forget the *nuance* behind ideas
- So… **write every idea down!**
  - Technology makes this easy nowadays
  - Desktop text file, Google Doc, iOS Notes app, etc.



You won't think of an idea twice!

- **You will forget ideas**
  - You will also forget the *nuance* behind ideas
- So… **write every idea down!**
  - Technology makes this easy nowadays
  - Desktop text file, Google Doc, iOS Notes app, etc.
- Go through your idea book periodically
  - Change in skills, interests, resources, news
  - Enhance, delete, and re-rank
  - **Junior researchers:** start documenting your ideas
  - **Senior researchers:** start organizing & grouping your ideas



Pushing Back on Ideas

- Questions to ask yourself:
  - What is the **fundamental problem** you are trying to solve?
  - What **key observations or insights** lead to your approach?
  - What exactly is **your approach** (in less than a paragraph)?
  - What must be built to implement the idea? **Can you do it?**
  - What are key **evaluation questions** to determine success?

Pushing Back on Ideas

- Questions to ask yourself:
  - What is the **fundamental problem** you are trying to solve?
  - What **key observations or insights** lead to your approach?
  - What exactly is **your approach** (in less than a paragraph)?
  - What must be built to implement the idea? **Can you do it?**
  - What are key **evaluation questions** to determine success?

- Design a **quick exploration** to "feel" for success
  - Did it work?
  - If not, why?

Gold

Pyrite (Fool's Gold)

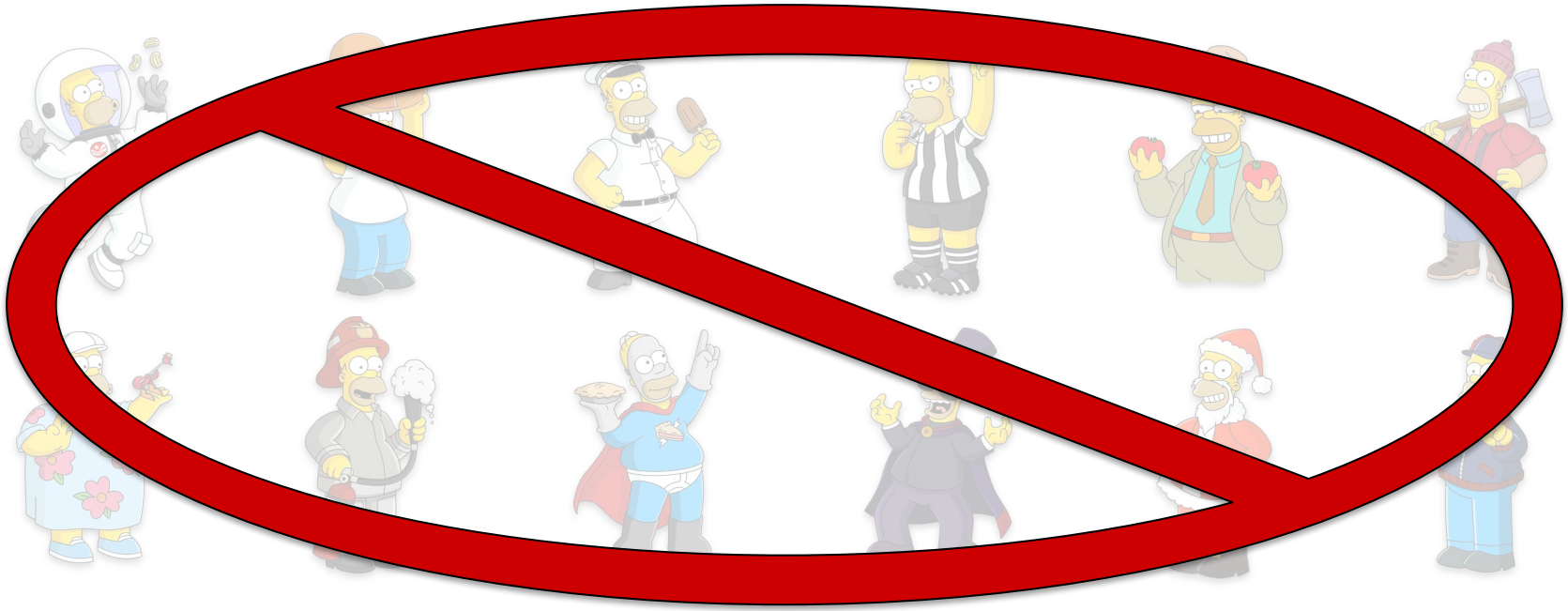
Pursuing Ideas

You have to make a hard choice...

What idea should I work on?

You have to make a hard choice...

What idea should I work on?



The Excitement / Doability Trade-off

- The best ideas are often **the most uncertain**
- Questions to ask yourself
  - Can I learn enough to do it?
  - Will this even work?
  - Will others find this interesting?
- Uncertain ideas often spawn future (more certain) work

Other Considerations

- Advisor (interests, funding)
- Department/lab resources
- Your skills and experience
- Available experts
- Potential collaborators
- How easy to publish/potential impact
- Publication venues

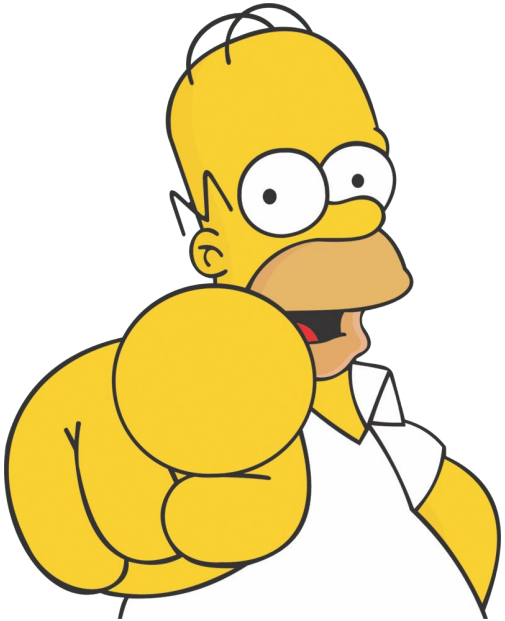
and...

The “First Mover” Advantage

- If you are not **the first** to an idea, you likely will face an **uphill battle**
- How far behind are you in **knowledge**, **resources**, and **infrastructure**?

Detouring vs. Committing

- Write distracting ideas down
- Talk with your advisor or labmates
- **Good ideas** can lead to **great ideas**
- **Be prepared to let an idea go**

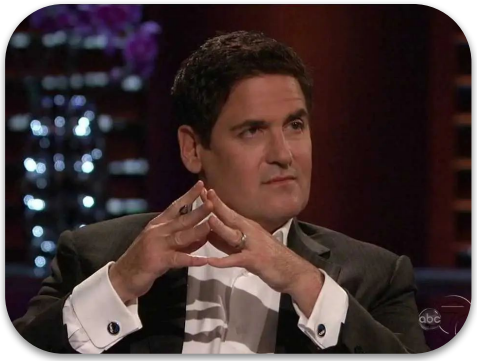


We don’t publish an idea...

Hey Mark! I’m seeking one trillion dollars so that I can build the world’s first flying car!

We don't publish an idea... we publish its proof!

Hey Mark! I'm seeking one trillion dollars so that I can build the world's first flying car!

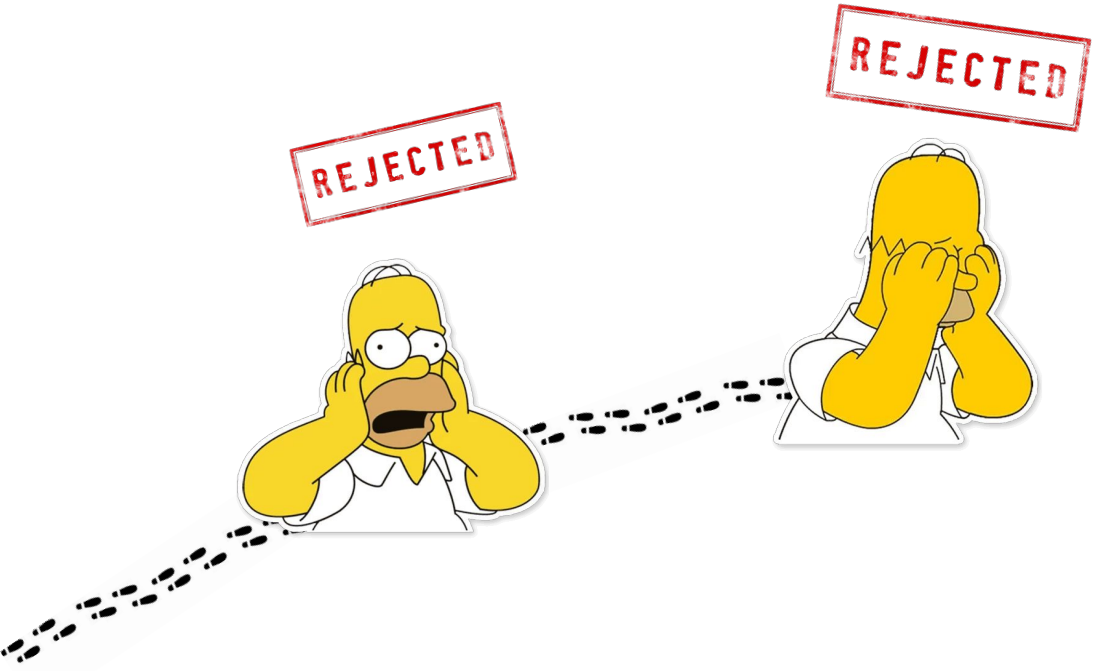


Hey Mark! I've built a flying car that costs **far less than a conventional car**, and I'd like one trillion dollars to mass-produce it...

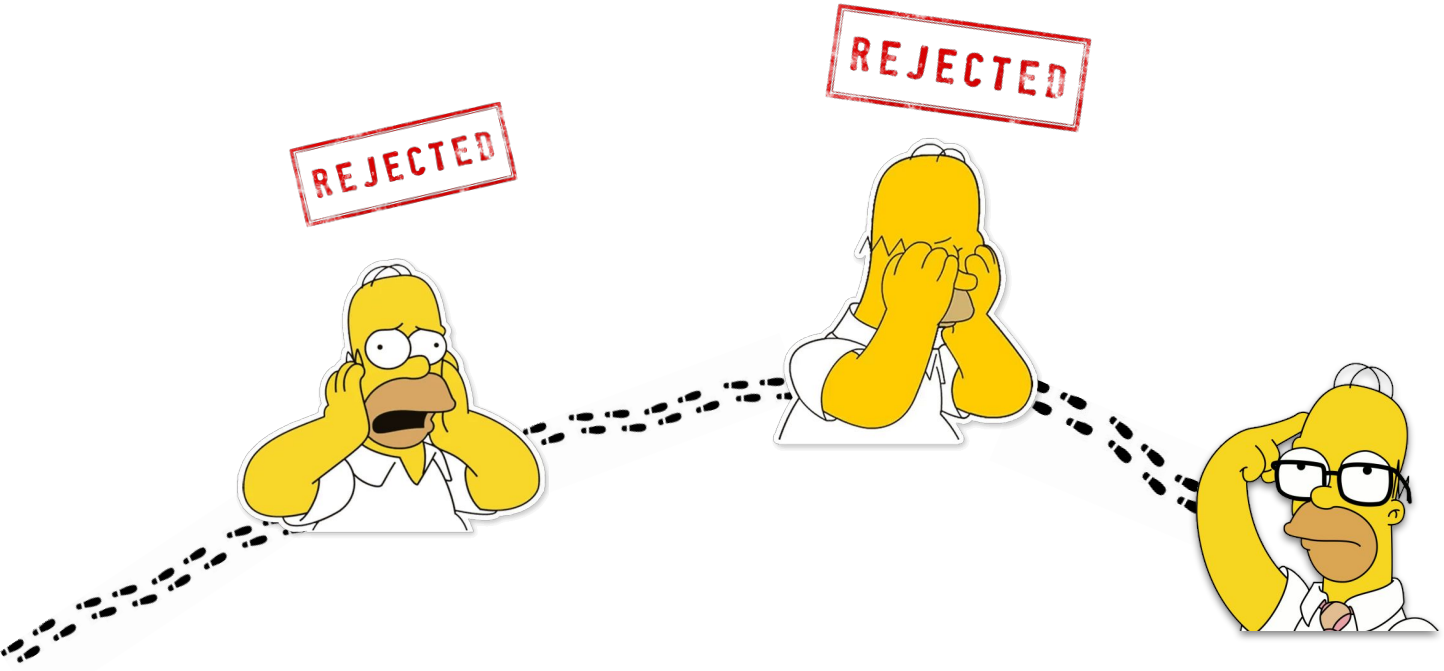
Research is a process...

Research is a random process...

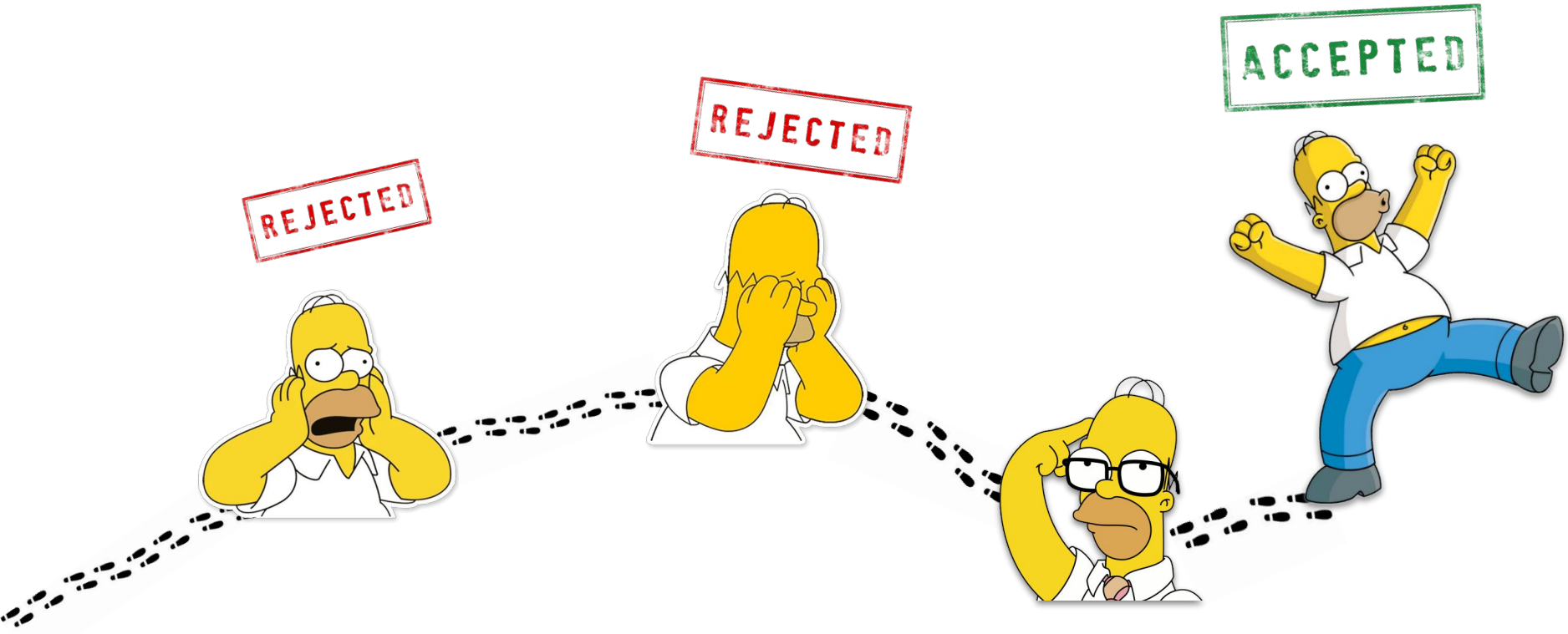
Research is a random process...



Research is a random process...



Research is a random *and* iterative process...



Resources

- **How to Look for Ideas in Computer Science Research** by Zhiyun Qian
- **Finding a Topic and Beginning Research** by Dianne Prost O'Leary
- **How to come up with research ideas?** on StackExchange
- **How to Get Startup Ideas** by Paul Graham
- **Creativity Techniques** on Wikipedia

Questions?

Next time on CS 5963/6963...

Research 101: Writing