Giving a Presentation

How to give a good presentation

School of Computing and Information

Department of Computer Science

How to give a good presentation

1. Decide on **content** of value to audience

2. **Organize** your thoughts on slides

3. Practice your **delivery**

How to give a good presentation

1. Decide on content of value to audience

2. Organize your thoughts on slides

3. Practice your delivery

Decide content based on audience and time

Your audience: Generally smart individuals

- Computer Scientists? Yes
- Knowledgeable about your area? Maybe
- Knowledgeable about your problem? Probably not

Time is usually limited

- Invited talk: < 1 hour
- Conference talk: 20 minutes or so
- Elevator talk: < 2 minutes
- Your talk: 15 minutes

This is not a lot of time...

Bottom line: Your audience should learn something from your talk

How to give a good presentation

1. Decide on content of value to audience

2. Organize your thoughts on slides
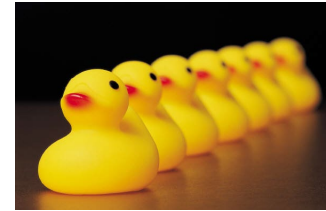
3. Practice your delivery

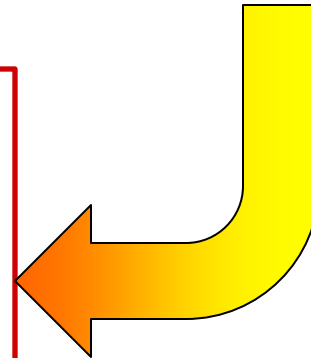# How should I organize my thoughts?

This is a hard problem…

… with interesting applications…

… that builds on prior work…

Two sub-parts:
- You solved a problem
- You used neat technological advancements to do this

Create an Outline

- It should be more than:
  - Introduction
  - Implementation
  - Outcome
- Write full **sentences** of what you want to get across
  - Intro: I worked on project X that fulfilled need Y
    - The project was challenging because it had to scale to a million users
    - Existing tools A, B, C are available to tackle the problem
  - Implementation: I used tool B to implement project X
    - Tool B was chosen over A and C because it is more scalable
    - I designed the system on the cloud to scale dynamically (diagram)
    - While coding, I found these difficulties with scaling and solved it by…
  - Outcome: Project X fulfilled need Y and more
- Putting the sentences together should form a **story**

Create Slides

- With a good outline, creating slides is straighforward

1. Title the slide based on the outline bullet point

2. Add slide content with focus on that title

3. Add figures in support of content

How to give a good presentation

1. Decide on content of value to audience

2. Organize your thoughts on slides
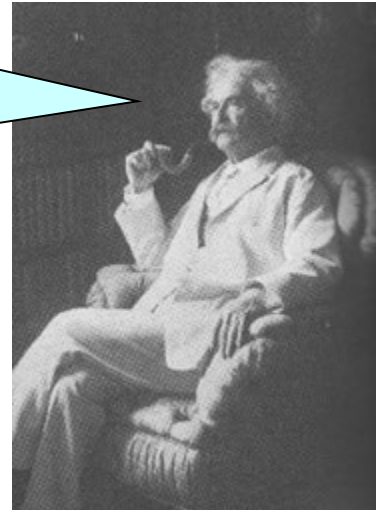
3. Practice your delivery

Practice, Practice, Practice

■ Practice makes better
- *Alone*: Work on your "script". Try recording yourself!
- *Peer group*: Get used to other people being around
- *Broader population*: Assess outsider comprehensibility

It takes three weeks to prepare a good ad-lib speech
- *Mark Twain*

It's not just *what* you say, but *how* you say it

- **Body language** says a lot
  - Make eye contact with your audience
  - Some movement is good to draw attention
  - Have a measured pace

- Present **one primary idea** per slide
  - Use slide titles to convey take-away message
- Refer to **every item** on the slide
  - If you don't, better to remove that item
- **Avoid reading** from your slides
  - Slides are there for the audience, not you

Make your delivery engaging

- Do not lose sight of the big picture
  - Audience should always know where you are taking them
  - Audience may need refocusing from time to time

- Give context
  - Why are you telling me this? Where does it fit in?
  - Why did you make that choice? What were the constraints?
  - Was that choice successful? Why or why not?

Pitfalls you want to avoid

University of Pittsburgh

Pitfall 1: Admire my beautiful slide

A slide is not abstract art
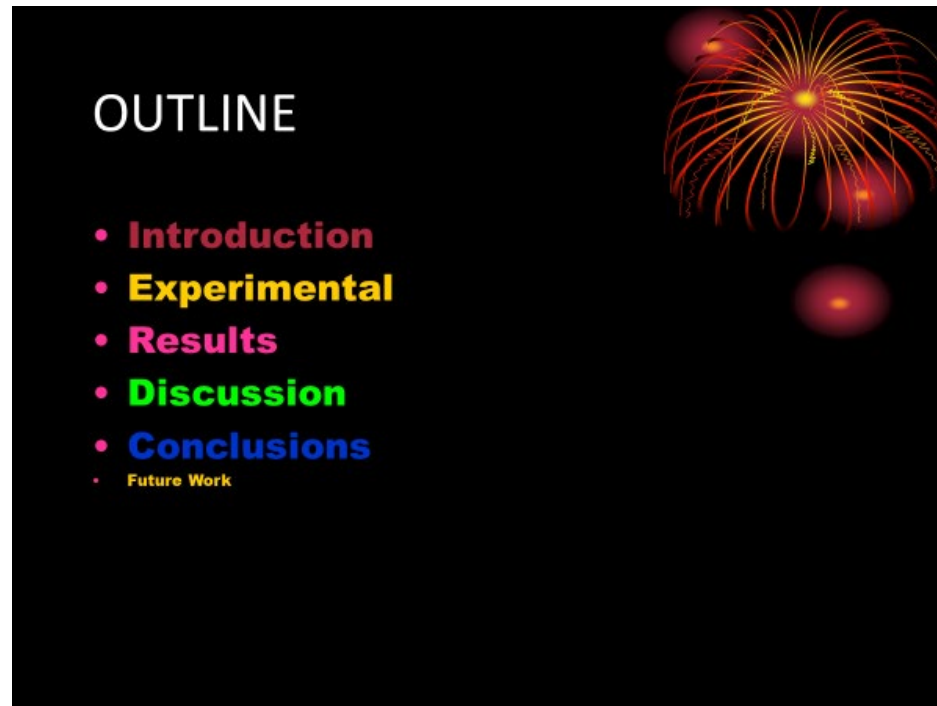
- **Fonts, colors, and style should be consistent**
  - If not, the difference should convey a meaning
- **By the way, was that outline slide *really* necessary?**
  - Most talks are structured that way – no information content

Pitfall 2: Look at my amazing code

■ This is my pseudocode for solving Hanoi towers:

```
void solve_hanoi(n, src -> dest, temp)
{
        if (n == 0) return;
        solve_hanoi(n-1, src -> temp, dest);
        move(1, src -> dest);
        solve_hanoi(n-1, temp -> dest, src);
}
```

↑ ↑
Warning: Example of a bad slide. Do not imitate.

Nobody wants to read your code

- If you still feel it is important

1. Explain at a **high level** what the code is doing

2. **Focus** audience attention at the interesting part

Towers of Hanoi: Recursive Solution

src   temp   dest
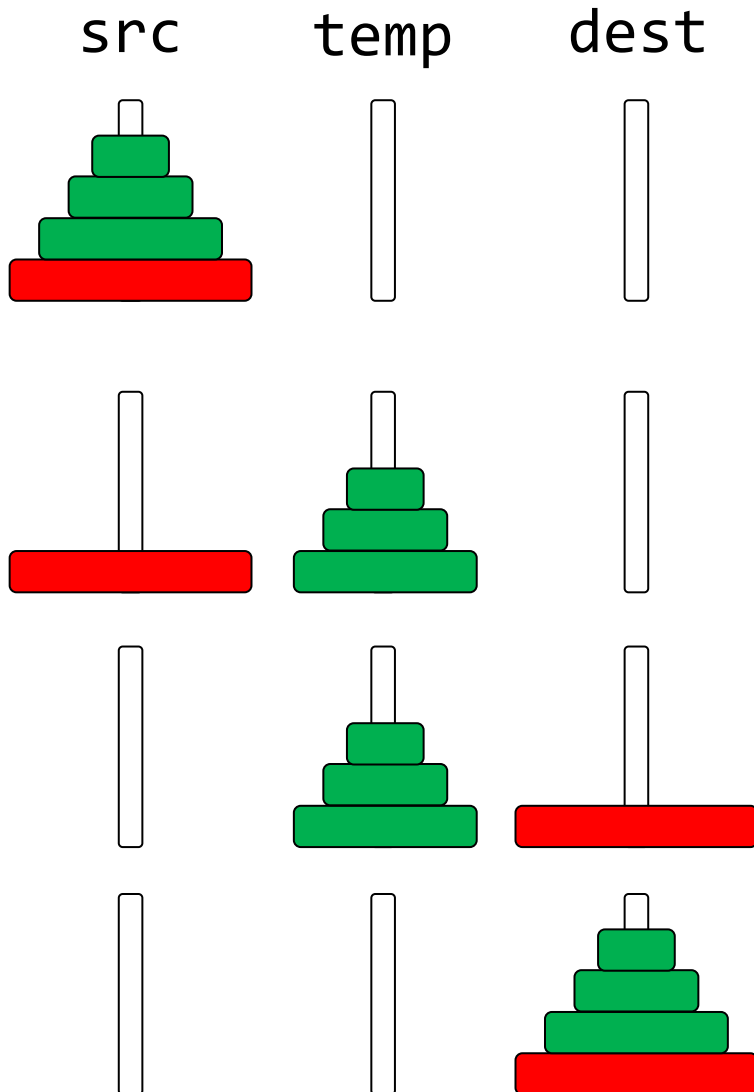
- **Move all disks at src → dest**
  - With the help of temp
  - Rule: disks must always be stacked smallest → largest

1. **Move n-1 disks into temp**
   - Recursive formulation as original problem, just with n-1 disks

2. **Move 1 disk into dest**

3. **Move n-1 disks into dest**
   - Again, n-1 version of same problem

Towers of Hanoi: Recursive Solution

■ Recursive solution for the Hanoi towers:

```
void solve_hanoi(n, src -> dest, temp)
{
    if (n == 0) return;
    solve_hanoi(n-1, src -> temp, dest);
    move(1, src -> dest);
    solve_hanoi(n-1, temp -> dest, src);
}
```

Solve moving n-1 disks with the power of recursion!

Pitfall 3: I am a math whiz

$$\text{score}(p, A.R, v) = \sum_{(C_i, w_i) \in \text{osets}_\omega(v.C, A.R)} w_i \cdot \frac{1}{2}^i$$

$$\omega_{len}(C_s, \_) = \gamma^{\max_{p \in \text{paths}(C_s)}(\text{length}(p))}$$

$$\omega_{ind}(C_s, C) = 1 - \frac{\max_{C_i \in C \setminus \{C_s\}}(|C_s \cap C_i|)}{|C_s|}$$

$$\omega_{li}(C_s, C) = \alpha \cdot \omega_{len}(C_s, \_) + \beta \cdot \omega_{ind}(C_s, C)$$

Well guess what. Many are not.

■ Translate math to plain English whenever you can

■ At least highlight what matters, and what is the take home message

$$\omega_{ind}(C_s, C) = 1 - \frac{\max_{C_i \in C \setminus \{C_s\}}(|C_s \cap C_i|)}{|C_s|}$$

Increasing the elements of C_s decreases the value of the function

- ## Proof sketch:

Monotonic. To prove the monotonicity of Equation 6, we proceed by induction. We first assume that principal p has previously discovered the (ordered) collection of proofs and weights $(C_1, w_1), \ldots, (C_n, w_n)$ for the role $A.R$. The base case that we must consider is that a new pair (C_s, w_s) is discovered such that no weight w_i is less than w_s. In this case, this new pair will introduce a new term to the end of the summation calculated by Equation 6, thereby increasing principal p's score for the role $A.R$.

Assume that (C_s, w_s) can be inserted before up to n terms in the sequence of (c_i, w_i) pairs while still preserving the monotonicity requirement. Now, assume that p has previously found proofs of authorization with the sequence of weights $S = (C_1, w_1), \ldots, (C_i, w_i), \ldots, (C_{i+n}, w_{i+n})$ and has now discovered a (C_s, w_s) pair such that $w_s > w_i$, thereby needing to be inserted before $n + 1$ terms in the sequence S. We first note that replacing (C_i, w_i) with (C_s, w) will generate a sequence S' that—when used in conjunction with Equation 6—will produce a score greater than that produced using S, since $w_s > w_i$ and all other terms are the same. By the inductive hypothesis, (C_i, w_i) can then be re-inserted before the n final terms of S' while still preserving monotonicity.

Then why am I listening to you?

- **Having too much to read can interfere with listening**

- **Reading and listening use same part of the brain**
  - Both require language processing

- **Use figures and diagrams as much as possible**
  - Interferes much less with listening
  - Write text as part of diagrams that visually add meaning
    - Tables, flow charts, tree structures, pyramids, …

- Don't leave a picture hanging there and expect your audience to interpret it

- If you have something on your slide that you don't explain, it is just **noise**.

A picture is worth a thousand words
But only if you explain it!

- **Put in graphical cues to focus attention**

- Point to the figure and explain each part

- Interpret the figure on behalf of the audience

Monthly outgoing mail workload, millions of units

October 1968   October 1970   October 1972

Election Months

Number of letters sent by congress
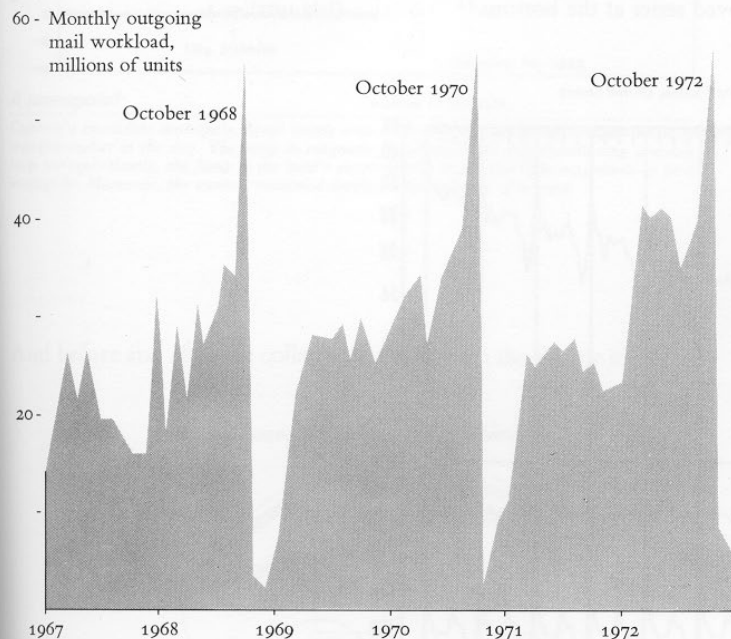
The graphic is worth at least 700 words, the number used in a news report describing how incumbent representatives exploit their free mailing privileges to advance their re-election campaigns:
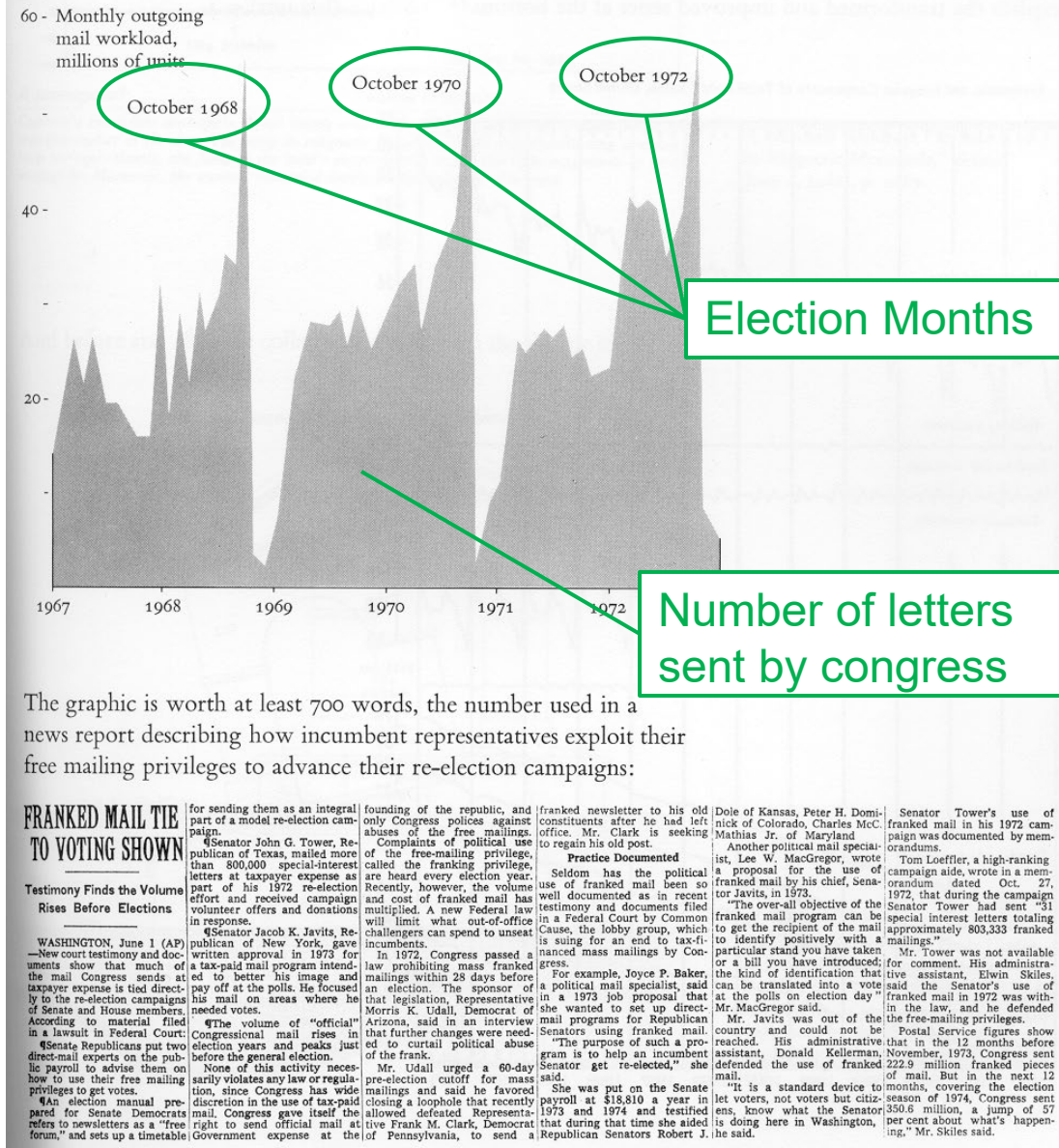
Pitfall 6: Acronyms and jargons makes me look smart!

- IMHO, ARE = ADIH. TBH, FUBAR & 2M2H.

- Translated: In my humble opinion, an acronym rich environment is another day in hell. To be honest. Its f***ed up beyond all recognition and too much to handle.
- Acronyms can be useful *but only if you explain them!*