Top Computer Science Skills As Valued by Industry

- Yes, programming skills are important (we need to know what we can do/not do, why, etc.)
- BUT given that, the most important skills are:
  - Problem solving
  - Communication
  - Adaptability
  - ... Team work
Being a Computer Scientist

- Solving problems
- Thinking algorithmically (how)
- Thinking critically (how well)
Today...

We are looking at:
• Ways to spark the **joy** of solving problems
  – Exercising this muscle
• Emphasizing **the journey**, not the result*
  – Deemphasizing failure (it happens)
• Providing **tools** to approach problem solving

*Contrary to what we discussed about inclusion, but not the same context 😊
Combination of:

- Try it yourselves
- Discuss the cases
- Tools (*see docs)
Warm-Up
Two fathers, each accompanied with their own son, go fishing. Each person catches a fish. However, together they only caught 3 fish. Why?
A bullet train leaves Paris at 230km/h, an hour later a train leaves Lyon at 150 km/h. When they cross, which train is closer to Paris?
So, What Did We Learn?

- **Short non-technical** problems
  - Quick turn around
  - No big deal whether correct or not, move on to the next problem and keep playing

- **Don’t give into your fear**
  - Nothing to show off
  - Keep cool and look for relevant clues
Let’s Keep Going a Little More... Patterns?
De Polignac Conjecture (1848)

Every odd integer > 1 is the sum of a prime $p$ and a power of 2.
What Do You Think?

- Does it work or not?
- Why?
What Could Go Wrong?

- Think of how your students may approach it
- What could be wrong about it?
How Would We Approach This Problem or Avoid an Incorrect Reasoning?

- What question(s) need to be asked?
- How to address them?
Morale of the Story

Beware of patterns
Hasty generalization
A Final Problem...

... Before we look at tools
Knight’s tour

On a chess board:

Can you find a path for a knight starting at the bottom left corner to more to the top right corner by visiting every cell on the board once and only once?
What Do You Think?

- Can it work or not?
- Why? How?
What Could Go Wrong?

- Think of how your students may approach it
- What could be wrong about it?
How Would We Approach This Problem or Avoid an Incorrect Reasoning?

- What question(s) need to be asked?
- How to address them?
So, What Did We Learn?

Be lazy, not busy
Ok, Let’s Get Concrete!

Tools
IDEAL, MIT 7 STEPS
IDEAL Framework for Getting Started

- **I** Identify the problem
- **D** Define and represent the problem
- **E** Explore possible strategies or solutions
- **A** Act on a selected strategy or solution
- **L** Look back and evaluate
IDEAL

- **I** Identify the problem:
  - What needs to be solved? What are we trying to?
- **D** Define and represent the problem:
  - How to formalize it? How do we represent it?
- **E** Explore possible strategies or solutions:
  - List a number of possible techniques that can be used / approaches
- **A** Act on a selected strategy or solution
  - Identify a ranking of the approaches / which one should be tried first?
- **L** Look back and evaluate
  - According to success / performance metrics
I + Heuristics

• Breaking down a problem into smaller pieces
• Including divide and conquer when relevant
Let’s Practice!

- The rickety footbridge
- The n-Queen problem
- Sudoku
Four people need to cross a rickety footbridge; they all begin on the same side. It is dark and they have on flashlight. A maximum of two people can cross the bridge at one time. Any party that crosses, either one or two people, must have the flashlight with them. The flashlight must be walked back and forth; it cannot be thrown, for example. Person 1 takes 1 minute to cross the bridge, Person 2 takes 2 minutes, Person 3 takes 5 minutes, and Person 4 takes 10 minutes. A pair must walk together at the rate of the slower person’s pace. For example, if person 1 and person 4 walk together, it will take them 10 minutes to cross the bridge. If person 4 returns the flashlight, a total of 20 minutes will have passed.

What is the minimum amount of time that they need to cross the bridge?
BONUS PROBLEM
Sudoku?
Challenge

Which one is harder? Designing a Sudoku solver or a Sudoku generator? How would you go about each?
Explain your Thoughts
Defend your Position