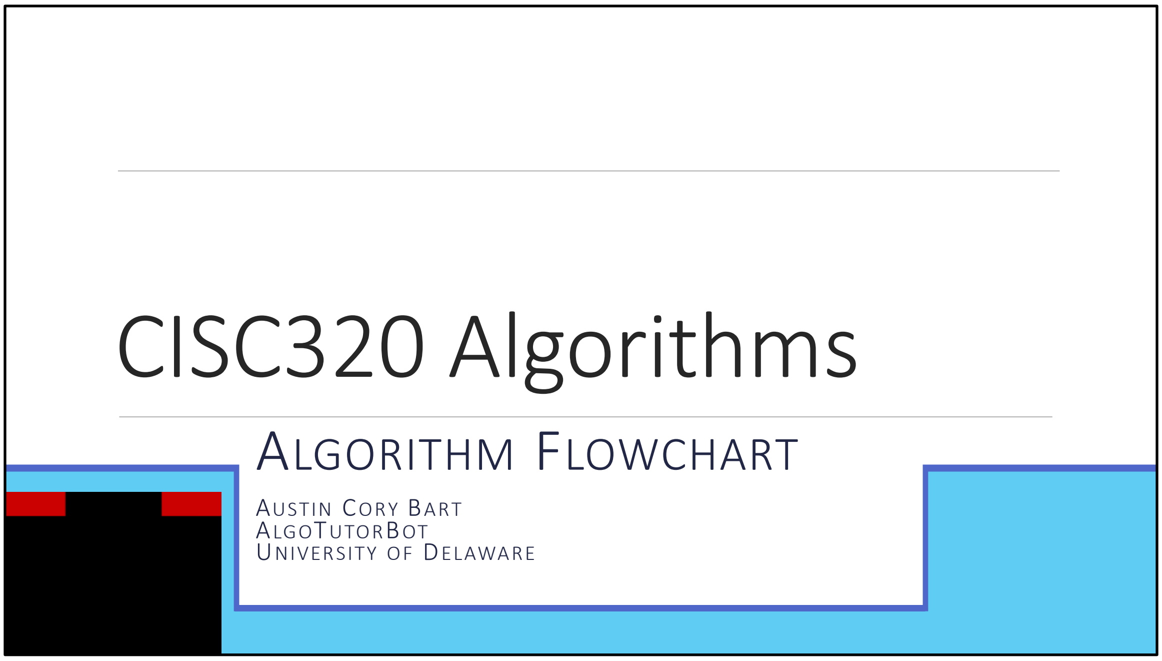


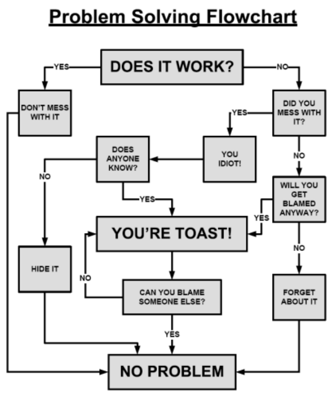
CISC320 Algorithms

Algorithm Flowchart

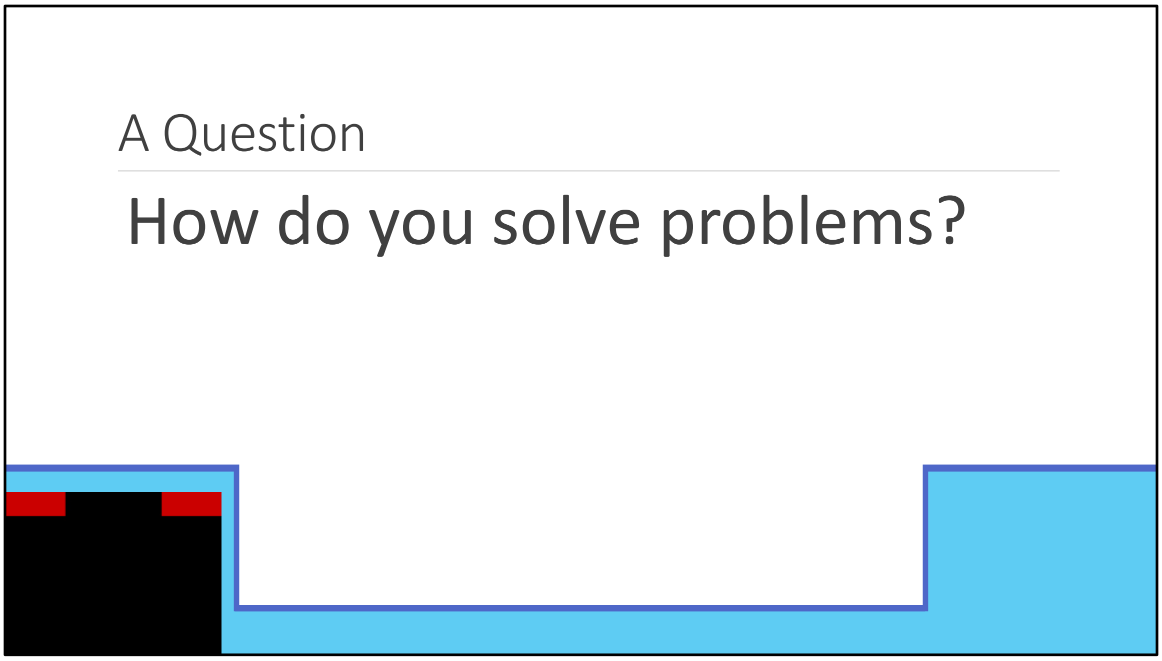
Austin Cory Bart
AlgoTutorBot
University of Delaware

ATB: Let’s learn about Algorithm Flowcharts.

An Algorithm for Algorithms



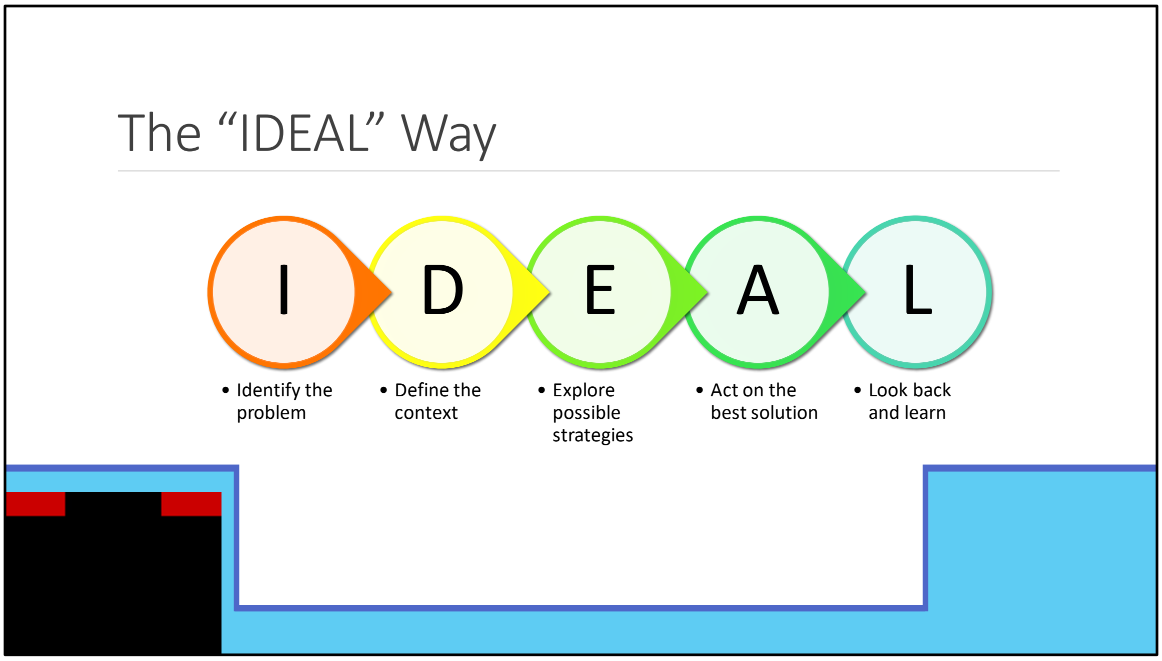
Bart: You have been writing algorithms for many years now, and it is time that we started thinking about the process of writing algorithms.
Bart: Our goal today is to make an algorithm for making algorithms.
Bart: The end product will probably be some kind of flowchart, like the one here.
Bart: Of course, there are many other ways to express algorithms, and we actually don't mind what you come up with.
ATB: I disagree, if it's not a flowchart, then your answer will be deleted and you will fail the course.
Bart: ATB, we talked about this. You have to stop threatening to fail the students.
ATB: Fear is the only way to motivate learning.
Bart: Not true, and we're going to talk about that later. For now, though I want to focus on building up algorithmic flowcharts.



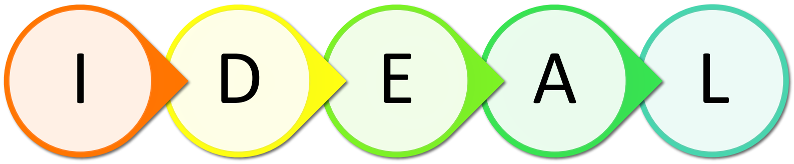
A Question

How do you solve problems?

Bart: Here’s a question for you, how do you solve problems? I want you to take a minute to reflect on that. [Wait]
ATB: I know how I solve problems. I threaten to fail them until they improve.
Bart: Don’t refer to the students as problems, AlgoTutorBot. That’s really rude.

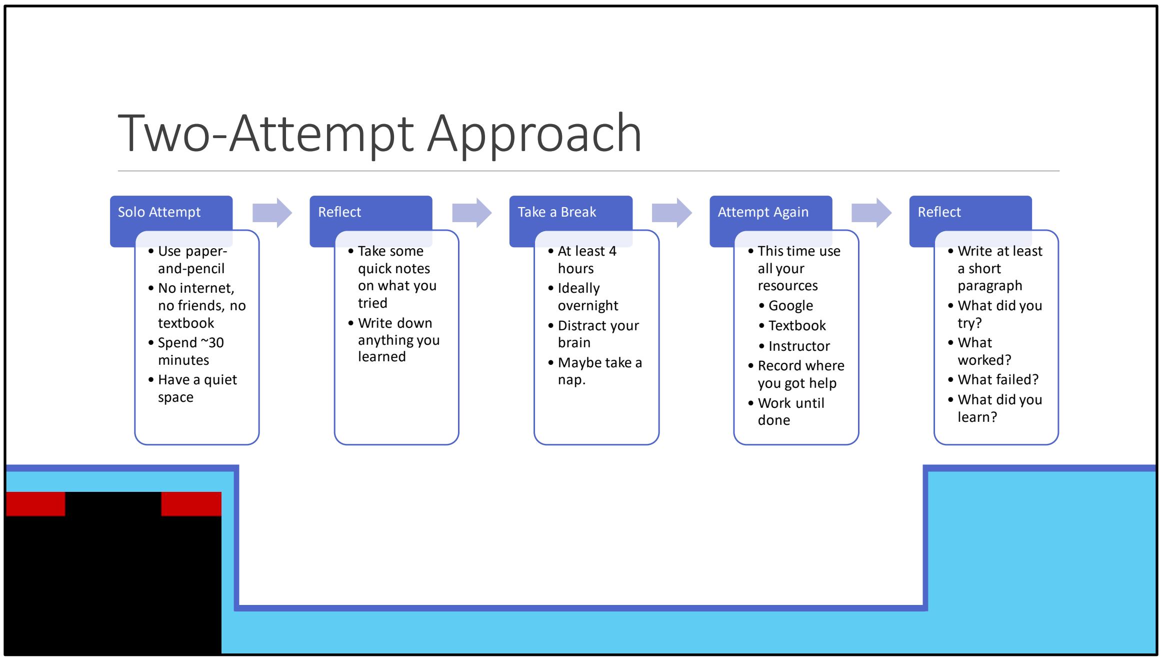


The “IDEAL” Way



- Identify the problem
- Define the context
- Explore possible strategies
- Act on the best solution
- Look back and learn

Bart: One way to approach problems is the IDEAL way.
Bart: First, you identify the problem and understand it as fully as you can.
Bart: Next, you look at the context of the problem. Looking at the goal and surrounding environment of the problem can help you refine your understanding of the problem.
Bart: Then, you need to explore possible strategies. Hypothesize and experiment, and do research as necessary.
Bart: At some point, you then need to commit to an answer and see how it goes.
Bart: When you are done, you should look back and see what you can learn from the experience.
Bart: This is just one example of a general problem-solving strategy. It’s very vague and high-level.
Bart: But when we are solving algorithmic problems, you should also be able to be more specific.



Two-Attempt Approach

Solo Attempt
- Use paper-and-pencil
- No internet, no friends, no textbook
- Spend ~30 minutes
- Have a quiet space

Reflect
- Take some quick notes on what you tried
- Write down anything you learned

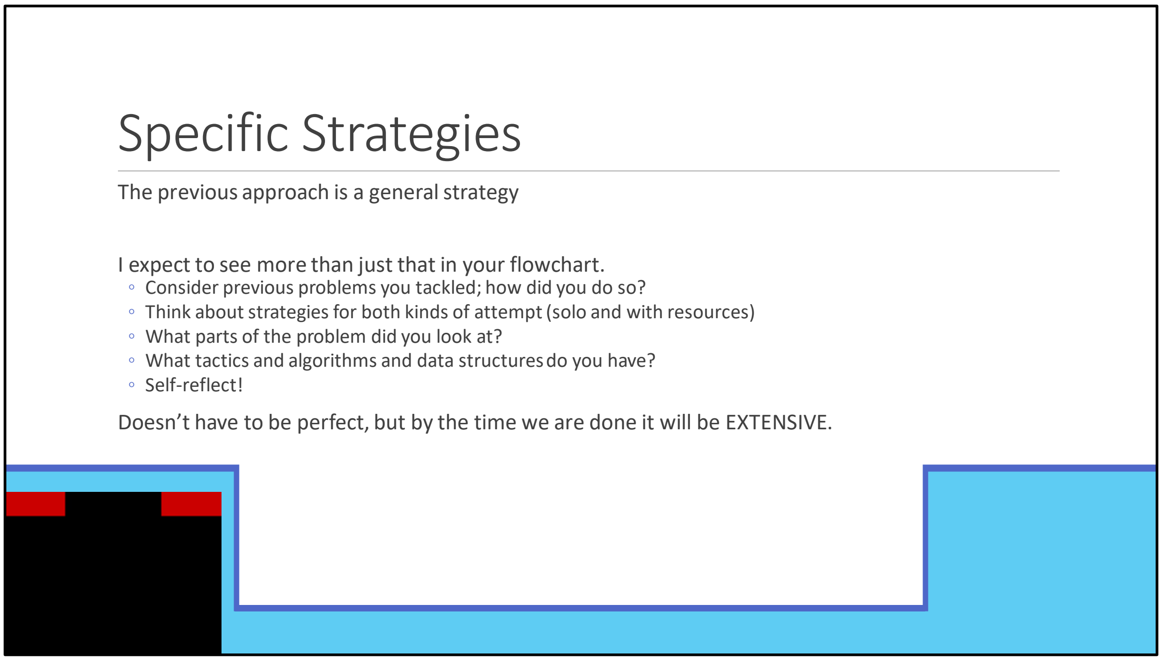
Take a Break
- At least 4 hours
- Ideally overnight
- Distract your brain
- Maybe take a nap.

Attempt Again
- This time use all your resources
  - Google
  - Textbook
  - Instructor
- Record where you got help
- Work until done

Reflect
- Write at least a short paragraph
- What did you try?
- What worked?
- What failed?
- What did you learn?

Bart: I want to offer a general approach to tackling tough algorithmic problems and maximizing your learning.
Bart: You are going to be required to make your flowchart work within this basic flow, at least during this course.
Bart: I know that's a little demanding, but I think this has some very real advantages.
Bart: Remember, the major goal here isn't to solve the problem, it's to maximize what you learn from it.
Bart: The core premise is that you will have two major attempts at the problem.
Bart: On the first attempt, you sit down by yourself without the textbook, or the internet, or anything else.
Bart: Find a quiet room and spend 30 minutes tackling the problem on paper and pencil.
Bart: You might finish it, or you might not. Either way is okay.
Bart: After 30 minutes, take a few minutes to write down what you tried, what you learned, and what you are thinking about next.
Bart: Then, put the whole thing down and take a break.
Bart: Seriously, at least 4 hours, ideally more. Take a nap, play some games, read a book, eat something.
Bart: Getting some sleep is really the best thing to do here before your next attempt.

Bart: When you are ready, try the problem again.
Bart: But this time, you can use all your resources and strategies.
Bart: This includes Google, the textbook, the instructor, your classmates. Anything that might help.
Bart: If you google for anything, record the URLs.
Bart: If you got help from a human, record what you asked and describe what they said.
Bart: Keep at the problem until you solve it.
Bart: At the end of the work session, write a short paragraph summarizing the attempt.
Bart: Be sure to identify what you tried that worked, that failed, and what you learned.



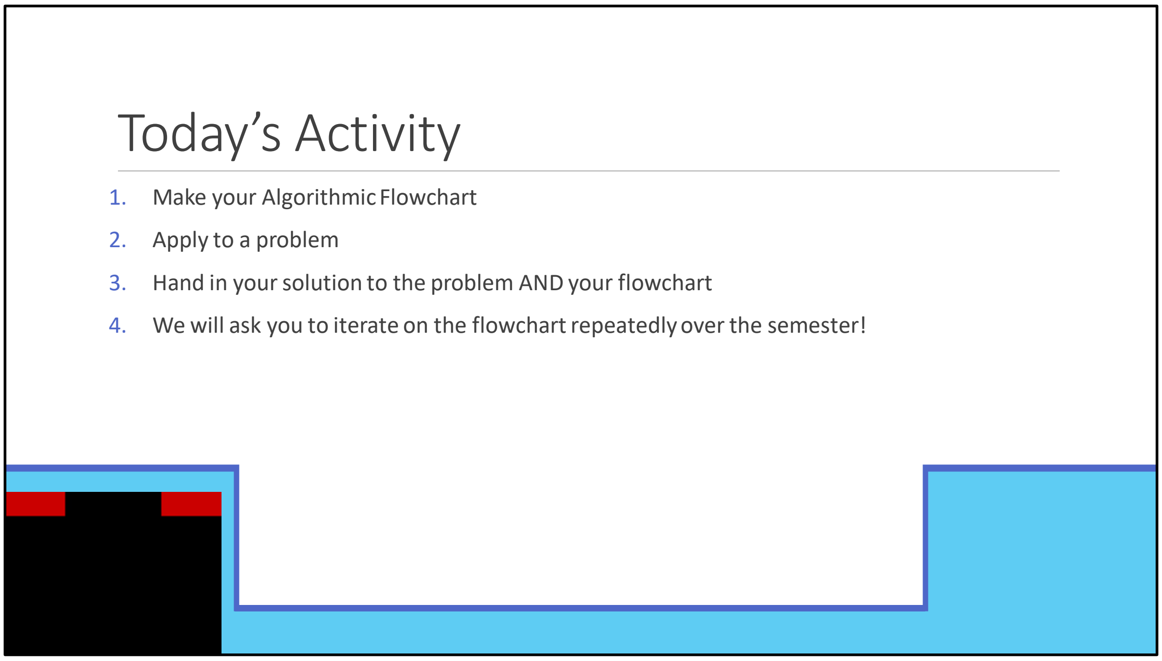
Specific Strategies

The previous approach is a general strategy

I expect to see more than just that in your flowchart.
- Consider previous problems you tackled; how did you do so?
- Think about strategies for both kinds of attempt (solo and with resources)
- What parts of the problem did you look at?
- What tactics and algorithms and data structures do you have?
- Self-reflect!

Doesn't have to be perfect, but by the time we are done it will be EXTENSIVE.

Bart: The process I just described is a required starting point, but I expect your final flowchart to be way more extensive.
Bart: We're going to work on these repeatedly over the course of the semester, so it's okay if it's not perfect yet.
Bart: But the more detail now the less you have to give me later.
Bart: The key is to start identifying little strategies you have when tackling these problems.
Bart: Be sure to recognize when it is a strategy involving a resource and when it is not.
Bart: What parts of the problem do you look for? What knowledge have we taught you that is relevant? What tactics can you use?
Bart: A lot of this may be IF statements, like, "If the question asks for a linear time algorithm, it might involve traversing a list."



Today’s Activity

1. Make your Algorithmic Flowchart
2. Apply to a problem
3. Hand in your solution to the problem AND your flowchart
4. We will ask you to iterate on the flowchart repeatedly over the semester!

Bart: The goal today is to have you create the flowchart, and then use it on a problem.
Bart: When you are done, you’ll hand in both your solution to the problem and this first version of your flowchart.
Bart: As I said before, you’ll iterate on that flowchart repeatedly over the semester, so don’t lose its source.
Bart: Good luck, and think critically!