Everyday Computational Thinking
It Can Save Lives

Paul S. Wang, Kent State University

February 23, 2020

Introduction

The great digital revolution is here and now. The Internet, the Web, and the computer in its many different forms, are providing instant communication across vast distances and changing nearly every aspect of our day-to-day living.

In the March 2017 edition of *AroundKent* (Vol 13, online at `aroundkent.net`), we gave an overview of *Computational Thinking* (CT) and stated "CT is thinking inspired by an understanding of IT, its advantages, limitations, and potential problems" and understood that CT was a powerful way of thinking.

We also defined a new verb **computize**:

> *To apply computational thinking. To view, consider, analyze, design, plan, work, and solve problems from a computational perspective. When considering, analyzing, designing, formulating, or devising a solution/answer to some specific problem, computizing becomes an important additional dimension of deliberation.*

In this article, we will explain how CT can be applied by everyone in everyday situations and how it can make important differences and even save lives.

What Is An Algorithm?

Remember the hugely successful movie *The Social Network* (2010)? It told the story of Mark Zuckerberg and how he created Facebook. The motion picture also introduced the term *algorithm* to much of the world for the first time.

From the author's book *From Computing to Computational Thinking* we understand an algorithm is a step-by-step procedure.

The origin of the word "algorithm" traces back to the surname Al-Khwārizmī of a Persian mathematician (780–850 CE), who was well-known for his work on algebra and arithmetic with Indian numbers (now known as Arabic numbers). The modern-day meaning of algorithm in mathematics and computer science relates to an effective step-by-step procedure to solve a given class of problems or to perform certain tasks or computations.

Specifically, a procedure becomes an algorithm if it satisfies all of the following criteria:

- **Finiteness**: The procedure consists of a finite number of steps and will always terminate in finite time.
- **Definiteness**: Each step is precisely, rigorously, and unambiguously specified.
- **Input**: The procedure receives certain data (or none) as input before it starts. Possible values for the data may vary within limitations.
- **Output**: The procedure produces results as its output.
- **Effectiveness**: Each operation in the procedure is basic and clearly doable.

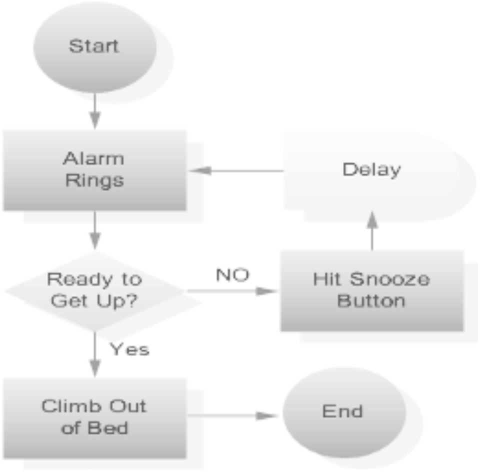
For a given problem, there usually are multiple algorithms for its solution. The design and analysis of algorithms are central to computer science and programming.

What have algorithms to do with everyday computational thinking? Good question. Well, it has to do with setting goals, devising concrete steps to achieve them, anticipating problems, and arranging solutions in advance. That's pretty important for everything everyday, right?

Flowcharts

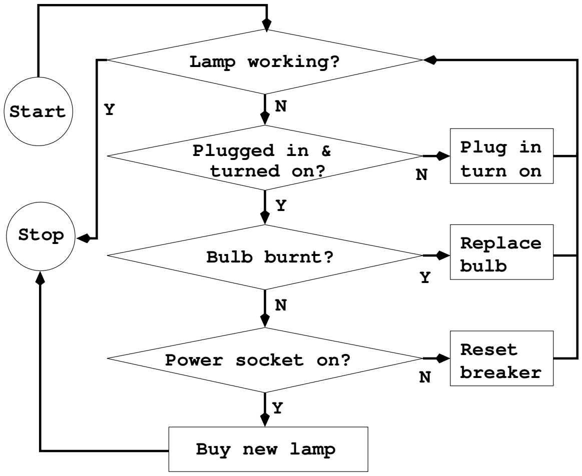
An algorithm is basically a procedure to achieve a certain goal.

A *flowchart* presents a procedure visually with words and diagrams. For any procedure, we can use a flowchart to plan the sequences of steps, to refine the solution logic, and to indicate how to handle different possibilities. Here is a simple flowchart for the task of "getting up in the morning."



We begin at the `Start` and follow the arrows to each next step. A diamond shape is used to indicate a fork in the path. Which way to turn depends on the conditions indicated. Obviously, we use diamond shapes to anticipate possibilities. The snooze option leads to a branch that repeats some steps. In programming, such a group of repeating steps is called a *loop*. The procedure ends when the person finally climbs out of bed.

As another example, let's look at a flowchart for troubleshooting a lamp.



The very first step after `Start` is significant. Although the purpose of the procedure is to troubleshoot a lamp, we nonetheless, make no implicit assumption that the lamp is not working. Without this step at the beginning, the procedure would potentially troubleshoot a perfectly good lamp, and, worse yet, would decide to replace it with a new lamp!

Each of the next three steps tests for a particular problem and makes a fix. Then the same procedure is reiterated by going back to step one to determine if the lamp is now working. This flowchart is a bit more complicated. Yet, it is worth careful examination. That is also a good way to get into the head of a programmer.

All you need is pencil and paper to start drawing your own flowcharts. Try it and you may find it not so difficult. To make nice looking flowcharts you can find many tools on your computer as well as online.

Correctly setting goals and anticipating potential problems are important aspects of devising a procedure.

We all were, and still are, horrified and outraged by the United Airlines passenger dragging incident (April 2017). But if the airline had set "passenger service" truly as its goal then it would have used a procedure that increased the incentives until getting enough volunteers to give up their seats. One can't help but wonder. Could this be the tip of an iceberg of problems caused by a wrong-headed culture? The incident unnecessarily inflicted much harm on the victim, the company, employees, shareholders and the police.

Everyday Applications

For us, we want to avoid fussy, vague, confused, wishful, emotional, impulsive, optimistic, or pessimistic thinking. We want to practice CT anytime and anywhere we can. We want to set clear goals, have a sequence of steps to achieve them, anticipate problems and prepare solution plans in advance.

Take driving a car for example. What is the goal? It is to get to a destination safely. It is not enjoying the sound system, watching the scenery, or engaging in conversation, although we have nothing against any of that as long as it does not get in the way of safe driving. Texting and driving is never safe.

Stopped at a traffic light, we wait for the light to turn green. But, we may need to run the red light if an 18-wheeler is about to crash into us from behind. That means we need to be checking our rear-view mirror while waiting for the green light. When the light turns green, do we blindly rush into the intersection? What if a car is running the tail end of the yellow light or the red light?

Thus, the goal is not to obey traffic signals, but to make sure it is safe. In the United States, a car crash kills a person every 12 minutes on average. If you are thinking straight, is a car a fun machine or a dangerous one? CT can keep you focused on the goals, make you pay attention to details, plan for contingencies, and shield you from distractions. CT can save the day, and perhaps even your life!

Now let's apply CT to the task of "getting ready to drive a car" and write down an algorithm-inspired pre-drive checklist.

1. Am I ready to leave? Forgot to bring anything?

2. Walk around the car, check windows, tires, lights, back seat, and any objects and activities near the car.

3. Get in the car, foot on brake, close and lock all doors.

4. Adjust seat and steering column positions as needed. Check positions of all rear-view mirrors, buckle up.

5. Check the instrumentation panel, pay attention to the fuel level.

6. If necessary, familiarize yourself with the controls for lights, turn signals, wipers, heat/AC, and emergency signal. Make sure they are working properly.

7. Release the hand break, start the engine, shift gear.

8. Make sure the gear is in D or R as intended, then start driving.

Airlines have developed rigorous preflight checklists for safety. Incidents, sometimes fatal, happen when pilots and crew, failing computational thinking, do not follow the exact procedure. The same goes for doctors and nurses in hospitals, especially in operating rooms.

You Can Do It!

We have given a sampling of computational thinking in everyday situations. However, we have by far, not exhausted the possibilities and will perhaps continue this series of articles on CT.

We hope you liked this article and please feel free to give your feedback directly to the author (`pwang@cs.kent.edu`).

For most people, CT does not come naturally. Research has shown that snap judgment based on intuition and experience is the norm. But, by keeping CT in mind and consciously making it an added dimension to our deliberations, we can make it work for us.

By being creative, everyone can derive benefits from CT every single day. As a result, our community, even the entire society, will be better off by becoming more efficient and effective.

Acknowledgments

Many thanks to Matt Keffer for the nice pictures and to Lonnie Hawks as well as Jennifer Wang for proofreading and helpful suggestions.