



Algorithms and templates 2

Outline

- In this lesson, we will:
 - Introduce the idea of algorithms
 - See that the function is the ideal mechanism for implementing algorithms
 - Introduce the idea of algorithm design techniques
 - Describe how we can generalize our algorithms for arbitrary data types

Algorithms and templates 3

To this point...

- Up to this point, we have focused on programming syntax
 - Basic syntax
 - Functions, conditional statements, for-loops and while loops
 - The structured programming theorem
 - Console input and output
 - Data types and binary
 - Arithmetic, comparison, logic, assignment and bit-wise operators
 - Arrays and pointers

Algorithms and templates 4

Algorithms

- A program is a sequence of instructions that has a processor perform a specified task
- An *algorithm* is a sequence of instructions that solves a specific problem
 - More formally:
 - An algorithm is a description of a process for solving a problem through a sequence of well-defined, precise and unambiguous instructions.
- An algorithm:
 - Has a well-defined set of requirements that must be met to allow the algorithm to successfully find a solution
 - Describes the format of the solution



Algorithms

- You have already seen a number of algorithms in both elementary and secondary school, even if you did not call them that
 - Finding the quotient and remainder of one integer m divided by another non-zero integer n
 - Using this, you can determine if one integer m is divisible by another integer n
 - Is the remainder zero?
 - How to determine if an integer n is a prime number
 - That is, does the integer, when divided by every integer between 2 and $n - 1$ have a non-zero remainder?
 - Find the prime factorization of an integer n
 - Find the greatest common divisor of two integers m and n
 - Using this, you can determine if two integers are *relatively prime*; that is, is the gcd equal to 1?



Algorithms

- Even in elementary school, you learned a number of algorithms:
 - How to add, subtract and multiply two numbers

$$\begin{array}{r}
 \overset{1}{5} \overset{1}{3} \overset{1}{2} \overset{1}{4} \\
 + \quad \overset{1}{9} \overset{1}{1} \overset{1}{7} \\
 \hline
 6 \overset{1}{2} \overset{1}{4} \overset{1}{1}
 \end{array}
 \qquad
 \begin{array}{r}
 \overset{5}{5} \overset{1}{6} \overset{1}{3} \overset{1}{0} \\
 - \quad \overset{3}{3} \overset{7}{7} \overset{7}{7} \\
 \hline
 5 \overset{2}{2} \overset{5}{5} \overset{3}{3}
 \end{array}
 \qquad
 \begin{array}{r}
 5 \overset{2}{2} \overset{3}{3} \overset{9}{9} \\
 \times \quad \overset{8}{8} \overset{1}{1} \overset{7}{7} \\
 \hline
 3 \overset{6}{6} \overset{6}{6} \overset{7}{7} \overset{3}{3} \\
 5 \overset{2}{2} \overset{3}{3} \overset{9}{9} \overset{0}{0} \\
 + 4 \overset{1}{1} \overset{9}{9} \overset{1}{1} \overset{2}{2} \overset{0}{0} \overset{0}{0} \\
 \hline
 4 \overset{2}{2} \overset{8}{8} \overset{0}{0} \overset{2}{2} \overset{6}{6} \overset{3}{3}
 \end{array}$$

- You also saw long division
- In secondary school, you saw polynomial addition, subtraction, multiplication and long division



Algorithms

- Some times, you may know an algorithm, but you may not understand why it solves the problem at hand
 - Here is an algorithm for calculating the square root of a number:

$$\begin{array}{r}
 \overset{2}{2} \overset{0}{0} \overset{9}{9} \\
 \sqrt{72361} \\
 \underline{-4} \\
 4 \overset{6}{6} \overset{3}{3} \overset{2}{2} \\
 \underline{-276} \\
 5 \overset{2}{2} \overset{9}{9} \quad 4761 \\
 \underline{-4761} \\
 0
 \end{array}$$



Algorithms

- We will find algorithms to solve the following problems on arrays:
 - Searching an array
 - Unsorted and sorted
 - Determining if an array is sorted
 - Sorting an array
- We will also look at an approach to solving problems:
 - Recursive algorithms:
 - An algorithm that takes a problem of a given size and
 - Breaks the problem into similar but simpler problems
 - Uses the same algorithm to solve those one or more simpler problems
 - Uses these solutions to the simpler problems to find a solution to the given problem
- Approaches to solving problems is referred to as *algorithm design techniques*





Summary

- Following this lesson, you now
 - Formally understand the idea of an algorithm
 - Understand you have learned a number of algorithms both in elementary and secondary school
 - Know that we will begin by looking at algorithms to solve problems on arrays, including searching and sorting
 - Know that we will follow this by a description of recursive algorithms and that we will look at how this algorithm design technique can be applied to finding algorithms that solve a number of problems



References

- [1] Wikipedia:
<https://en.wikipedia.org/wiki/Algorithms>



Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using Consolas.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

<https://www.rbg.ca/>

for more information.



Disclaimer

These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.

