Exercise: Algorithms

Each week, exercises will have some activities that can be done in groups. This week's activity will introduce you to the process. We'll use a simple game to explore basic searching algorithms.

Model 1 Hi-Lo Game

Hi-Lo is a number guessing game with simple rules, played by school children.

a) There are two players – *A* and *B*.
b) Player *A* thinks of a number from 1 to 100.
c) Player *B* guesses a number.
d) Player *A* responds with "too high", "too low", or "you win".
e) Players *B* and *A* continue to guess and respond until *B* wins (or gives up).

1. How many different answers can player *A* give?

2. When does the game end?

3. Play the game a few times to ensure that everyone understands the rules.

4. Identify 4–5 different guessing strategies that Player *B* could use. Each strategy should describe a **different approach** to the game. For example: *Start at 1, and count up until the correct answer is found.* In computer science, we call such strategies **algorithms**. Try to have a mixture of simple and clever algorithms, including ones that young children could use.

a)

b)

c)

d)

e)

5. Rank order the algorithms with regard to how **fast** they will find the right answer. Write 1 for the fastest algorithm (fewest guesses) and 5 for the slowest one (most guesses).

6. Rank order the algorithms with regard to how **easy** they are to describe or specify. (Suppose you had to explain them to a first-grader so that he/she could play the game.) Write 1 for the algorithm that is easiest to describe and 5 for the one that is hardest.

In computing, we often must search for a particular item in a set. Computer scientists are particularly interested in searching very large sets, with thousands or millions of values. For example, the Harvard University Library has roughly 16,000,000 volumes, and the US Library of Congress has roughly 22 million cataloged books and over 100,000,000 total items.

Exercise: Algorithms

Complete the following Chapter Review Problems (not in the textbook).

1. Euclidean algorithm. Euclid wrote one of the first known algorithms around 300 BC, and it's still in use today! Refer to Figure 0.2 (on Page 3) of the textbook to find the GCD of the following numbers. Use the provided tables to show the values of M, N, and R after each step. There may be more rows than needed. **Circle or * the final answer in each table.**

12 and 18		
M	**N**	**R**

1000 and 5040		
M	**N**	**R**

21 and 34		
M	**N**	**R**

34 and 55		
M	**N**	**R**

3. Name that chapter. Use the Table of Contents and Section 0.3 to match the topics on the right with their corresponding chapters on the left. Each chapter is referenced only one time.

1. Data Storage	_____	how algorithms are represented and discovered
2. Data Manipulation	_____	what problems can/cannot be solved, and why
3. Operating Systems	_____	how data is organized on disk; relational model
4. The Internet	_____	the layer between hardware and applications
5. Algorithms	_____	the mathematics of modeling and rendering
6. Programming Languages	_____	how data is organized in computer memory
7. Software Engineering	_____	CS meets psychology, biology, and linguistics
8. Data Abstractions	_____	different paradigms; compilers vs interpreters
9. Database Systems	_____	1's and 0's, logic gates, and digital circuits
10. Computer Graphics	_____	software life cycle, large software projects
11. Artificial Intelligence	_____	how computers are connected to each other
12. Theory of Computation	_____	machine language and program execution