What Is Computer Science?

Many people, if asked to define the field of computer science, would likely say that it is about programming computers. Although programming is certainly a primary activity of computer science, programming languages and computers are only tools. What computer science is fundamentally about is computational problem solving—that is, solving problems by the use of computation.

This description of computer science provides a succinct definition of the field. However, it does not convey its tremendous breadth and diversity. There are various areas of study in computer science including software engineering (the design and implementation of large software systems), database management, computer networks, computer graphics, computer simulation, data mining, information security, programming language design, systems programming, computer architecture, human–computer interaction, robotics, and artificial intelligence, among others.

Part I: The Essence of Computational Problem Solving

In order to solve a problem computationally, two things are needed: a representation that captures all the relevant aspects of the problem, and an algorithm that solves the problem by use of the representation. Let's consider a problem known as the *Man, Cabbage, Goat, Wolf Problem*.

A man lives on the east side of a river. He wishes to bring a cabbage, a goat, and a wolf to a village on the west side of the river to sell. However, his boat is only big enough to hold himself, and either the cabbage, goat, or wolf. In addition, the man cannot leave the goat alone with the cabbage because the goat will eat the cabbage, and he cannot leave the wolf alone with the goat because the wolf will eat the goat.

1. How does the man solve his problem?

There is a simple algorithmic approach for solving this problem by simply trying all possible combinations of items that may be rowed back and forth across the river. Trying all possible solutions to a given problem is referred to as a **brute-force approach**.

What would be an appropriate representation for this problem? Since only the relevant aspects of the problem need to be represented, all the irrelevant details can be omitted. A representation that leaves out details of what is being represented is a form of **abstraction**.

The use of abstraction is prevalent in computer science. In this case, is the color of the boat relevant? The width of the river? The name of the man? No, the only relevant information is where each item is at each step. The collective location of each item, in this case, refers to the state of the problem. Thus, the start state of the problem can be represented as follows.

man	cabbage	goat	wolf
[E,	**E,**	**E,**	**E]**

In this representation, the symbol E denotes that each corresponding object is on the east side of the river. If the man were to row the goat across with him, for example, then the representation of the new problem state would be

man	cabbage	goat	wolf
[W,	**E,**	**W,**	**E]**

in which the symbol W indicates that the corresponding object is on the west side of the river—in this case, the man and goat. (The locations of the cabbage and wolf are left unchanged.) A solution to this problem is a sequence of steps that converts the initial state,

[E, E, E, E]

in which all objects are on the east side of the river, to the goal state,

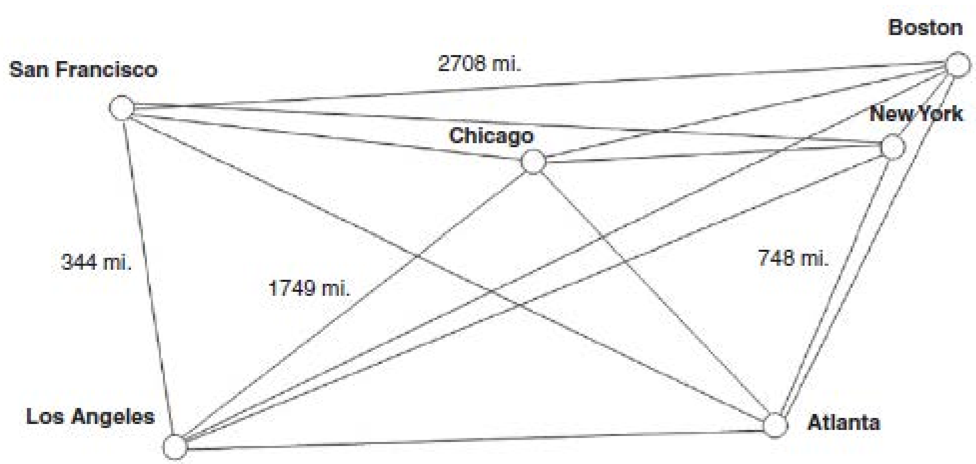
[W, W, W, W]

in which all objects are on the west side of the river. Each step corresponds to the man rowing a particular object across the river (or the man rowing alone).

As another example computational problem, suppose that you needed to write a program that displays a calendar month for any given month and year, as shown at the left. The representation of this problem is rather straightforward. Only a few values need to be maintained—the month and year, the number of days in each month, the names of the days of the week, and the day of the week that the first day of the month falls on. Most of these values are either provided by the user (such as the month and year) or easily determined (such as the number of days in a given month). The less obvious part of this problem is how to determine the day of the week that a given date falls on. You would need an algorithm that can compute this. Thus, no matter how well you may know a given programming language or how good a programmer you may be, without such an algorithm you could not solve this problem.

Part II: Limits of Computational Problem Solving

Once an algorithm for solving a given problem is developed or found, an important question is, "Can a solution to the problem be found in a reasonable amount of time?" If not, then the particular algorithm is of limited practical use.



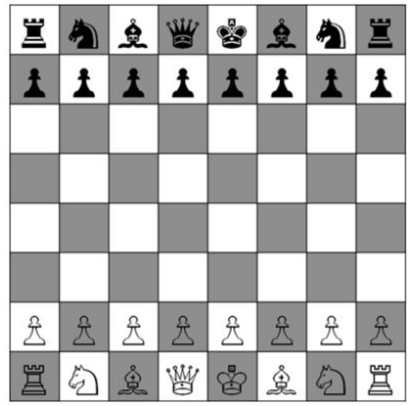
The *Traveling Salesman problem* (above) is a classic computational problem in computer science. The problem is to find the shortest route of travel for a salesman needing to visit a given set of cities. In a brute force approach, the lengths of all possible routes would be calculated and compared to find the shortest one. For ten cities, the number of possible routes is 10! (10 factorial or $10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 3{,}628{,}800$).

2. For twenty cities, the number of possible routes is 20! Calculate 20!.

If we assume that a computer could compute the lengths of one million routes per second, it would take over 77,000 years to find the shortest route for twenty cities by this approach.

For 50 cities, the number of possible routes is over 1064. In this case, it would take more time to solve than the age of the universe!

A similar problem exists for the game of chess (Figure 1-6). A brute force approach for a chess-playing program would be to “look ahead” to all the eventual outcomes of every move that can be made in deciding each next move. There are approximately 10120 possible chess games that can be played. This is related to the average number of look-ahead steps needed for deciding each move.



For problems such as this and the Traveling Salesman problem in which a brute-force approach is impractical to use, more efficient problem-solving methods must be discovered that find either an exact or an approximate solution to the problem.

Concepts and Procedures

1. A good definition of computer science is “the science of programming computers.” (TRUE/FALSE)

2. Which of the following areas of study are included within the field of computer science?
    a. Software engineering
    b. Database management
    c. Information security
    d. All of the above

3. In order to computationally solve a problem, two things are needed: a representation of the problem, and an   that solves it.
    a. Leaving out detail in a given representation is a form of   .
    b. A “brute-force” approach for solving a given problem is to:
    c. Try all possible algorithms for solving the problem.
    d. Try all possible solutions for solving the problem.
    e. Try various representations of the problem.
    f. All of the above

4. For which of the following problems is a brute-force approach practical to use?
    a. Man, Cabbage, Goat, Wolf problem
    b. Traveling Salesman problem

    c. Chess-playing program
    d. All of the above

5. Calculate 13! (13 factorial).

Problem Solving

1. Search online for two areas of computer science other than those mentioned in the introduction. Write a brief summary of each.
   .
2. For the Man, Cabbage, Goat, Wolf problem:
    a. List all the invalid states for this problem, that is, in which the goat is left alone with the cabbage, or the wolf is left alone with the goat.

    b. Give the shortest sequence of steps that solves the MCGW problem.

    c. Give the sequence of state representations that correspond to your solution starting with (E,E,E,E) and ending with (W,W,W,W).

    d. There is an alternate means of representing states. Rather than a sequence representation, a set representation can be used. In this representation, if an item is on the east side of the river, its symbol is in the set, and if on the west side, the symbol is not in the set as shown below,
        i. {M,C,G,W}—all items on east side of river (start state)
        ii. {C,W}—cabbage and wolf on east side of river, man and goat on west side
        iii. { }—all items on the west side of the river (goal state)

        Give the sequence of states for your solution to the problem using this new state representation.

    e. How many shortest solutions are there for this problem?

3. For a simple game that starts with five stones, in which each player can pick up either one or two stones, the person picking up the last stone being the loser,
    a. Give a state representation appropriate for this problem.

    b. Give the start state and goal state for this problem.

    c. Give a sequence of states in which the first player wins the game.