Should Students Always Use Algorithms To Solve Problems?

David V. Frank
Department of Physical Sciences, Ferris State College, Big Rapids, MI 49307
Claire A. Baker
College of Education, Butler University, Indianapolis, IN 46208
J. Dudley Herron
Chemistry Department, Purdue University, W. Lafayette, IN 47907

As chemistry teachers, we ask our students to solve both exercises and problems. Students do not see the difference and approach both the same way—by looking for an algorithm that fits their interpretation of the question. For example, students run into difficulty if they do not recognize when an algorithm needs to be modified. Such students don't think of problem solving as "what you do when you don't know what to do" (*1*). When we teach problem solving, we need to help students apply algorithms as a *part* of the problem-solving process, not as the entire means of solution.

Exercises versus Problems

John Hayes (*2*) has defined problem solving as a process that involves two conditions. First, there must be a gap between where you are now and where you want to be. Second, you cannot see an obvious way of crossing that gap. If the task you are working on meets both criteria, then you are solving a problem.

In school, we also expect students to solve other tasks called "exercises". The solution of an exercise usually involves the identification and correct application of an algorithm (*3*). On the other hand, the use of an algorithm is never sufficient for solving a problem. A problem solver often has to transform the problem into a recognizable form before he can even begin to identify which algorithms might be helpful.

Students who do not recognize the difference between exercises and problems are too eager to use algorithms. For example, consider students trying to solve this problem: "The molecular ion XF_2^- has three pairs of nonbonding electrons around the central atom. What is the bond angle around X?" Many students immediately categorize this as a molecular geometry problem. This categorization is an aid for some of these students and a straitjacket for others. The problem solvers focus on the critical attribute of the problem (five pairs of electrons around the central atom), visualize the structure, and determine the bond angle from the structure. The exercise solvers are derailed in a number of ways. Some students believe that they must write a Lewis structure before they can solve this problem. They stop when they cannot determine what group X is in, even though that is an unnecessary step. Other students, instead of trying to visualize the structure, recall only the name of the structure. Many of the students are subsequently unable to associate the correct bond angle with the name.

Based on a paper presented at the 190th National Meeting of the American Chemical Society, Chicago, IL, September 9, 1985.

As this example illustrates, algorithms aren't necessarily bad. Successful solvers use algorithms after they have decided what they need to do. They modify familiar solutions to fit their present problem. Unsuccessful problem solvers try to match previous solutions to the current problem; they don't change the algorithms that they use.

Teaching Algorithms and Problem Solving

One of the biggest difficulties in teaching problem solving is that many of the tasks that are problems for our students are only exercises for us. If we present our solutions in lecture format only, we may not successfully communicate how to handle the difficult steps. One way around this difficulty is to solve problems *with* our students, instead of solving problems *for* them. We can ask our students what steps we should be doing, instead of running through a well-practiced script. Teachers who use this strategy can make students aware that they are not expected to know the entire solution *before* they even start to solve the problem. Otherwise, weak students may not realize that every problem solver has to struggle.

Many algorithms are useful shortcuts that will work much of the time (for exercises), but may actually prevent understanding when a student finds a real problem. For example, how many of us have told students that the first step in stoichiometry is to write a balanced chemical equation? It is not always so. Some problems, such as determining the mass of magnesium oxide that forms when 10 g of magnesium is burned in air, can be solved without the use of a chemical equation. Likewise, many textbooks state that writing a Lewis structure is the first step in solving a molecular geometry problem. This obscures the main goal: determining the number of regions of high electron density around the central atom. We would be better off if we suggested that equations and Lewis structures are useful but not necessary tools for solving problems.

Sometimes we have a tendency to teach too many algorithms. Students often encounter this terminology when they are learning stoichiometry: mole–mole problems, mass–mass problems, mass–volume problems, and limiting-reagent problems. They learn a separate algorithm for each type, and they never see how these problems are essentially related. Instead, we should teach all of these as stoichiometry problems.

Finally, our choice of exercises may lead students to develop inflexible algorithms. Grouping homework problems together under a specific heading, such as "Charles's Law Problems", eliminates a lot of the prerequisite thinking that students need to do. Giving too many problems of the same type can also be harmful. For instance, the students who tried to determine the group of X in the molecular geometry problem had previously worked several exercises in which they were specifically asked to determine an unknown group. They apparently had developed mindsets that told them they had to determine X any time they didn't know its group.

If we want students to be able to solve novel problems outside the classroom, we must give them help. In addition to helping students use algorithms, we should also teach general problem-solving strategies (or heuristics). We can ask students "What is the goal of this problem?" or "How can we form a model of this problem?" Books by Polya (*4*), Wickelgren (*5*), or Whimbey and Lochhead (*6*) illustrate general problem solving at a level that high school or college students can understand.

Should students always use algorithms to solve problems? Of course not! If a student's first response is to decide which algorithms to use, then he or she is not solving a problem at all. However, when a student applies an algorithm as one step, modifies an algorithm, or even creates a new algorithm, then he or she is making use of algorithms as an efficient tool for unlocking problems.

Literature Cited

1. Wheatley, G. H. MEPS Technical Report No. 84.01, 1984, Purdue University, W. Lafayette, IN.
2. Hayes, J. R. *The Complete Problem Solver*; Franklin Institute: Philadelphia, 1980.
3. Kantowski, M. G. *J. Res. Math. Educ.* **1977,** *8,* 163–180.
4. Polya, G. *How To Solve It*; Princeton University: Princeton, NJ, 1957.
5. Wickelgren, W. A. *How To Solve Problems*; W. H. Freeman: San Francisco, 1974.
6. Whimbey, A.; Lochhead, J. *Problem Solving and Comprehension*; Franklin Institute: Philadelphia, 1982.