Algorithms — Before or After?

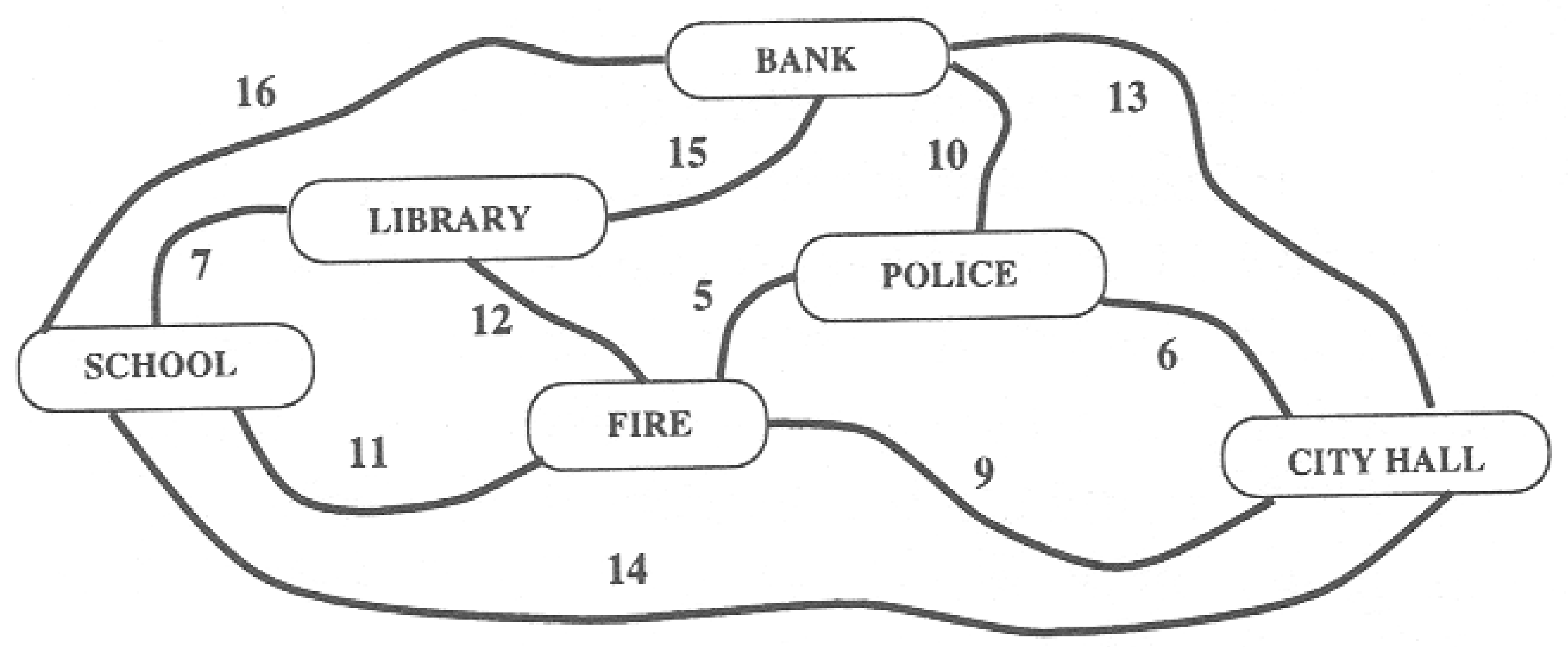
by Carol Price

I am a mathematics teacher at Robert E. Lee High School, an urban and ethnically diverse school in Baton Rouge, Louisiana. I teach Algebra 2 and Advanced Math (both regular and honors versions) to mostly 10 - 12th graders. One day, upon committing the sin of missing school for a math workshop, I left a poor unsuspecting substitute teacher to start a unit on weighted graphs. I prepared the students for my absence by letting them know that they would be starting another "discrete math" unit. That was all that they needed to hear—they loved the first unit I gave them on graph coloring and felt confident about their ability to handle anything labeled "discrete."

I left the substitute armed with numerous worksheets on minimum spanning tree problems. The directions for this first sheet were to connect all of a set of cities (directly or indirectly) to minimize the cost. The students were also told to avoid cycles because they waste money. That was it. They were left to contemplate other problems involving installation of water pipes, electrical wires, telephone wires, roads, etc., with only the help of their partner. The students had never heard the term *minimum spanning tree*, and they were sent off to battle without even a hint of an algorithm for protection.

I am a firm believer in "not giving the students what they don't need." If I tell the students how to solve a problem, not only am I telling them that they can't do it without me, I am taking away their chance to feel mathematically powerful. On the other hand, I was not sure how far the students would be able to go in this case.

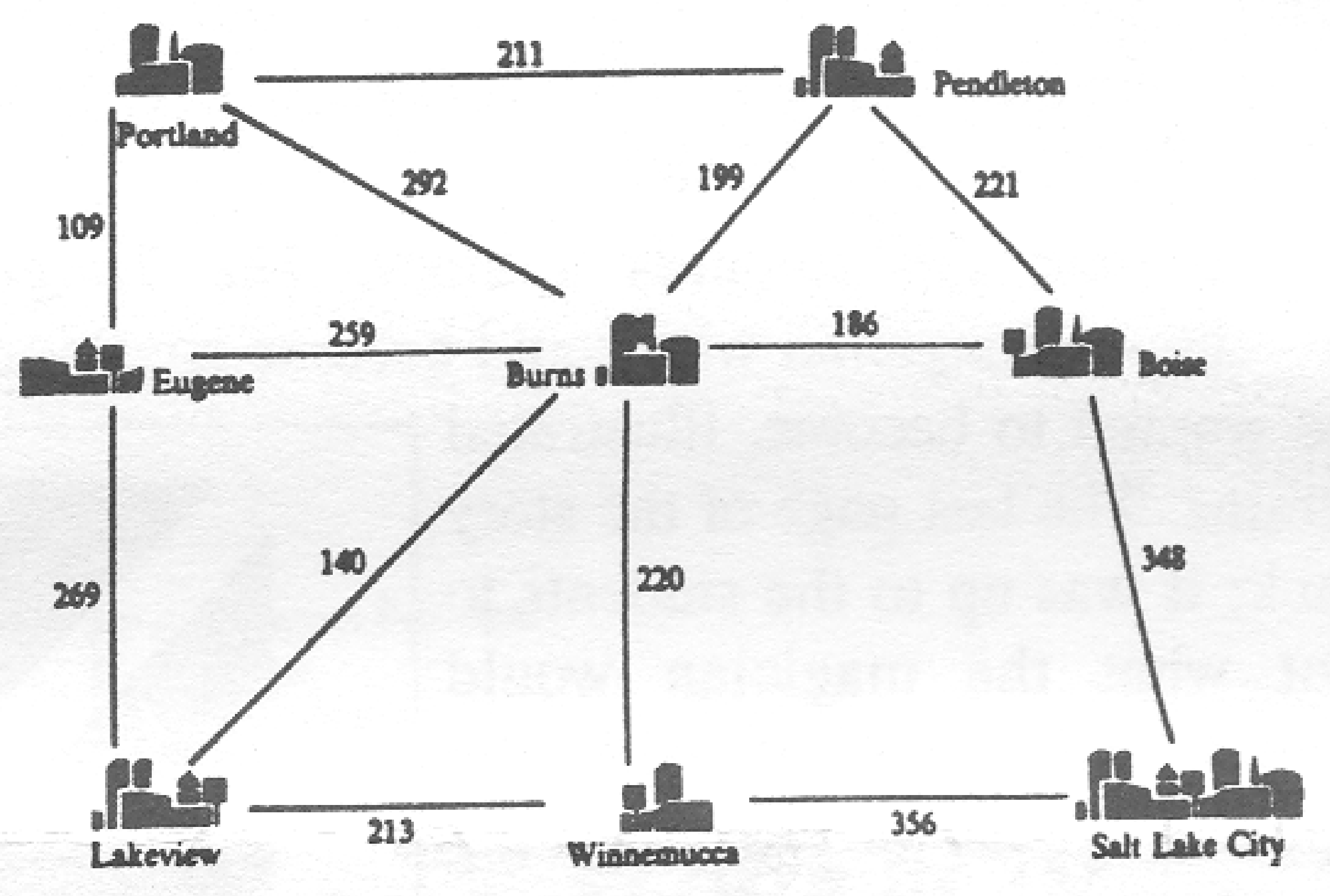
But, I was overwhelmed when I read the substitute's report. The majority of students breezed through worksheet after worksheet, and even helped the ones that were struggling. Almost everyone solved the problems correctly!



A Minimum Spanning-Tree Problem
The Muddy City Council wants to pave roads joining the six major buildings at minimum cost. The cost of paving each of the possible roads is given (in $100K); which roads should be paved? [3]

The next morning, I praised the students for their outstanding behavior and performance, and handed back their papers. It was at that time that I introduced Kruskal's algorithm[1] ([1], pp. 43-47) and Prim's algorithm ([2], pp. 529-530), both of which the students had discovered on their own. (A problem on water pipes led the students right to Prim's algorithm because they realized that the pipes had to stay connected.)

Having mastered the minimum spanning-tree problem, we moved on to the shortest-path problem. Again, I sent them diving into the problem without so much as a hint.



A Shortest-Path Problem
A family wants to drive from Portland to Salt Lake City. Distances are given in miles—what is the shortest route they can take? [4]

Their only directions were to get from *A* to *B* using the shortest path possible. As I walked around the room, I saw a few groups jotting down paths and comparing their lengths, some listing paths with tree diagrams, and a few actually discovering Dijkstra's algorithm ([2], Sec. 13.2). Of course, the latter two approaches generated a shortest path; the first approach generated a response more quickly, but did not always give the shortest path. After having some of the students explain their approaches and answers to the class, and discussing the advantages and disadvantages of their methods, I then introduced Dijkstra's algorithm.

On the third and final day of our weighted graphs unit, we discussed the famous Traveling Salesperson Problem (TSP). For the final time, I sent them diving into the problem without any help. After discussing their various approaches and solutions, we discussed algorithms and computer solutions. They were stunned that there currently isn't any algorithm that can solve this solve this type of problem quickly for a large number of cities. One could hear loud

(Continued on page 9)

[1]In the spirit of this article, no algorithms are described here! References are given, however.