## Discretion...

(Continued from page 5)

the consumer and non-traditional algebra students, I discussed multiples instead.

The second part of the lesson brought another code to the students' attention. While examining their textbooks they discovered the ISBN (International Standard Book Number). The students were fascinated by the breakdown by number group: group identifier, publisher identifier, title identifier, check digit. (I also explained that the bigger publishers have small publisher-identifier numbers to allow the title identifier to be longer. This led to a discussion of the number of possible titles available given a certain number of digits in the title identifier group.) The check digit in the ISBN number works like the zip code check, except that the sum is taken mod 11 instead of mod 10. By examining several ISBN numbers, the students were able to figure this out on their own.

The calculus and advanced mathematics students accomplished these discoveries in one class period, although a second period was necessary to fully discuss modular arithmetic and equivalence. The consumer and algebra students were only able to handle the zip code on the first day: they needed quite a bit of guided practice to interpret the bar code. I gave them a worksheet with addresses on one side and the bar codes on the other and asked them to match them up. I also encouraged my students to do some exploration on their own. The section on codes in *The Mathematical Tourist* by Ivars Peterson [3] is accessible to interested students.

As it turned out, coding methods were not what I taught in this lesson. Nor did I teach counting methods. Though I did not expect it, the end result of the lesson, rather than an understanding of codes, was a true appreciation of algorithms. Although several of my students have studied computer science, only a few could explain that an algorithm is a "step-by-step procedure." I wheedled an understanding out of each class through examples: "I know you all know an algorithm for dressing each morning, or for starting the car. You learned an algorithm in third grade for long division." As a result the word has become part of my students' vocabularies.

FURTHER READING

1. Gallian, Joseph, "How Computers Can Read and Correct ID Numbers," *Math Horizons,* Winter 1993, 14-15.

2. Lefton, Phyllis, "Number Theory and Public-Key Cryptography," *Mathematics Teacher,* 84 (January), 54-63.

3. Peterson, Ivars, *The Mathematical Tourist,* New York: W.H. Freeman and Company, 1988.

**How can errors be found in zip codes?**

You may wonder why the particular set of 5-digit blocks were chosen to represent the digits 0-9 for the zip code. In fact, this system allows one to detect errors **within** the blocks. Notice that each block of five has exactly two 1's and three 0's. Thus, if any one bar is misread you can tell that an error has been made. Observe, however, that you might not be able to tell where the error occurred: if 11001 is read, the original could have been 11000 (0), 01001 (4), or even 10001 (7). Also, if **two** bars are misread, you might not even notice: if the correct block is 10100 (9), and you misread the first two digits, you would think that the number was 01100 (6). Thus, this is a "single-error-detecting code", but is NOT a "single-error-*correcting* code". Now suppose that a single error occurs in one of the blocks. The final check digit actually allows you to **correct** the error. Figure out how this works by trying the following example:(solution on p. 6).

You might also try to explain how you can sometimes correct **two** errors this way. For more information and other references, see:

Malkevitch, Joseph, "Have you seen ... (Codes)" and "Mini-Bibliography...Codes", *In Discrete Mathematics,* Issue 3 (Aug. 93), p. 1, 8-9.

COMAP, *For All Practical Purposes,* Freeman, 1994, Chapter 9.

## *Solving the TSP...*

(Continued from page 3)

(This is an example of a "linear integer programming problem.") What makes the problem hard is that the number of constraints grows exponentially with the number of vertices. The strategy in using "cutting planes" is as follows. First find a vector which satisfies only **some** of the constraints, but does minimize the sum; this will be an approximate solution. Then, add new constraints (called "cutting planes") one at a time, so that each time you compute a new approximate solution it is closer to the true solution. To picture this, imagine that you want to find a diamond that has been baked into a cake, and that you're only allowed to make straight cuts. Certainly the method is effective only if the number of cutting planes needed does not grow too large. Sophisticated mathematical methods from linear algebra are needed both to find the approximate solutions and compute the cutting planes. So far, no one has found a better method for solving very large TS problems exactly.