PROBLEM-SOLVING WITH THE MICRO-COMPUTER

ONG SIT TUI

Computers and Mathematics

In his article, "Computer-Assisted Solutions to Numerical Puzzles", Tan (1983) highlighted the power of computers in solving mathematical problems. Indeed, with the help of the computer, Appel and Haken (1977) of the University of Illinois had proved in their article, "The Solution of the Four Color-Map Problem", that four colours will suffice for any planar map so that no two adjacent countries are in the same colour. They essentially reduced the problem to about 1930 sub-problems each of which could be routinely solved by a computer. The computer took 1,200 CPU hours and made an estimated 10 billion individual computations.

The purpose of this article is to show how a certain type of mathematical problem can be solved by a computer following a simple iterative procedure.

Algorithmic Thinking

Computers solve problems by following predetermined procedures. One such procedure is the *algorithm* which is a precise and unambiguous method for solving a specified problem in a finite number of steps. Utilizing the amazing high-speed computational capabilities of the computer, algorithmic procedures can be used to enable us to solve complex problems by a systematic and exhaustive search of the most appropriate solutions. For instance, the application of *Euclidean Algorithm* to determine the greatest common divisor of two integers is one of the many mathematical experiences of all students in secondary schools. The formulation of the most general and effective algorithms for the solutions to problems is one of the important tasks in mathematics.

What follows is a discussion of the "five sailors and a monkey problem" to illustrate how algorithmic thinking can be effectively used in problem-solving. Let us restate the problem briefly:

> Five sailors gathered a pile of coconuts to be divided equally the next morning. During the night, a sailor awoke and separated the coconuts into five equal piles with one left over which he gave to a monkey. He hid one pile and put the rest back together. He was followed in turn by the other four sailors, each of whom did exactly the same thing. Next morning, the remaining coconuts were divided equally with one left over for the monkey. Find the least number of coconuts the sailors could have begun with.

Mathematical Thinking

Tan (1983:32) reduced the above problem to the following Diophantine equation with two unknowns:

$$1{,}024N = 15{,}625F + 11{,}529$$

where N is the original number of coconuts and F the number each sailor received at the final division in the following morning. A BASIC program was written to search all possible integers from 1 to 2,000. The least solutions obtained were F = 1,023 and N = 15,621.

Incidentally, we can express N in terms of F as follows:

$$N = 15F + 11 + \frac{265(F+1)}{1024}$$

Since 265 and 1024 are co-prime, i.e. they have no common factors, 1024 must divide (F+1) to obtain an integral N. Obviously the least value of F is 1023. N can be solved by substitution.

An elegant algebraic solution to a more general problem for any number of sailors was given by Brashear (1967:599):

$$N = a^{a+1} - (a-1)$$

where a is the number of sailors. When a=5, the formula gives the value N = 15,621.

Algorithmic Strategies

A variety of algorithmic strategies can be formulated to solve the "five sailors" problem without much algebraic manipulation. These algorithms are recursive search strategies based on computation and exhaustive consideration of all possibilities within a specified range of solutions. Two diametrically different algorithms are discussed below to illustrate the distinct characteristics of algorithmic thinking.

(1) Forward Algorithm

The *forward algorithm* follows the natural development of the problem. If all sailors were honest, there should be at least 6 coconuts, i.e. N=6. Let R denote the number of coconuts left over after each theft. They started with N=R. Then there were 4(R−1)/5 coconuts left in the pile after each sailor had taken his share. If we repeat this process 6 times, including the final partitioning in the morning, we should always obtain integral solutions for 4(R−1)/5. Figure 1 illustrates this forward strategy of finding N.

The following BASIC program was written to run this algorithm on an APPLE II PLUS micro-computer. The solution was obtained in less than 2 minutes.

```
  5 REM —— FORWARD ALGORITHM ——
 10 FOR N = 6 TO 20000 STEP 5
 20 R = N
 30 FOR I = 1 TO 6
 40 R = 4 * (R − 1) / 5
 50   IF R < > INT (R) THEN 100
 60 NEXT I
 70 PRINT "LEAST NUMBER OF COCONUTS = "; N
 80 PRINT "EACH SAILOR'S FINAL SHARE = ";.R / 4
 90 STOP
100 NEXT N
110 END

RUN
LEAST NUMBER OF COCONUTS = 15621
EACH SAILOR'S FINAL SHARE = 1023

BREAK IN 90
```

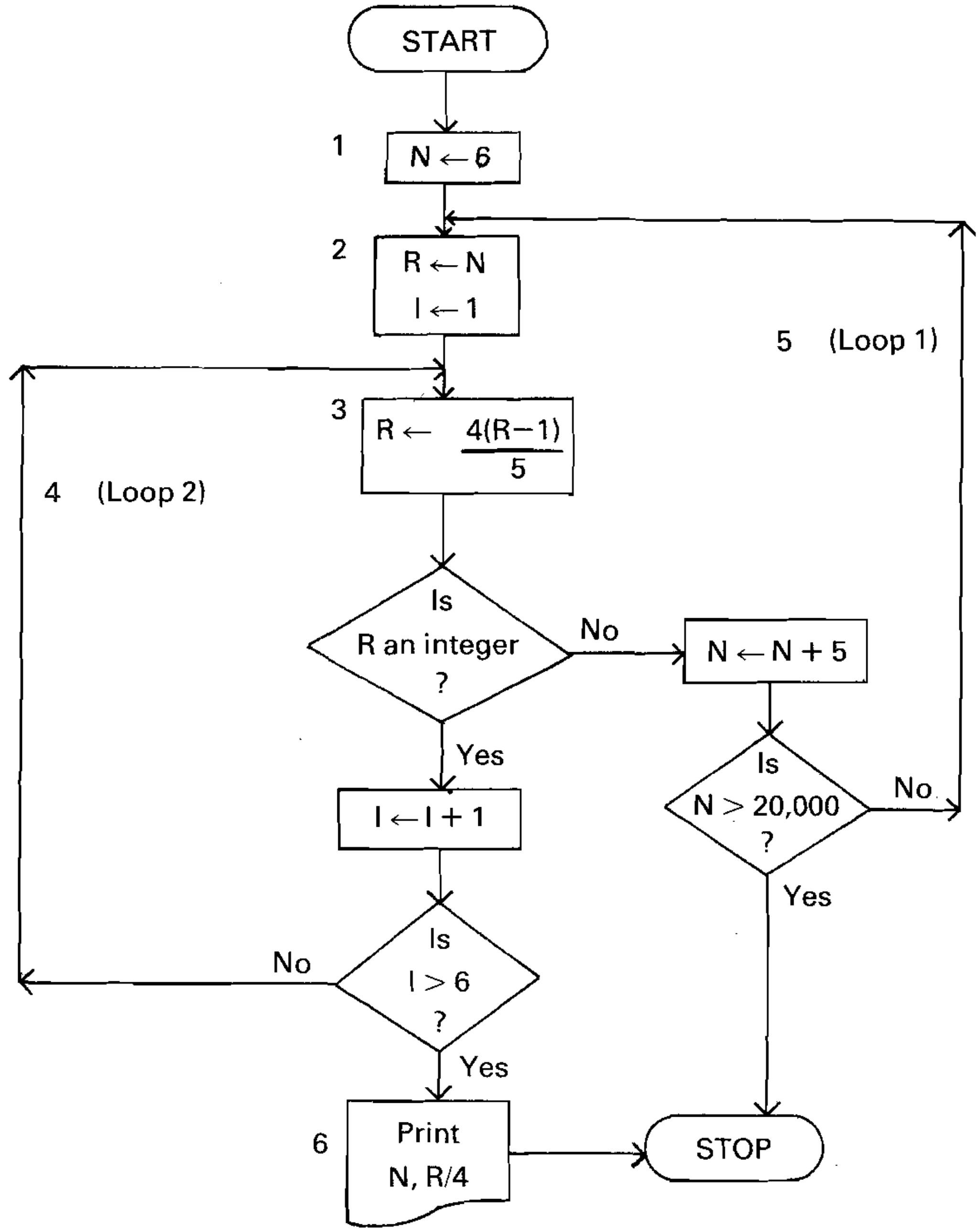

Figure 1. A flowchart for determining the number of coconuts using *forward algorithm.*

Flowchart Notes

1 There should be at least 6 coconuts, i.e. N=6.

2 N is the counter for Loop 1 and I for Loop 2. Initially, the number of coconuts left over, R=N, before the first theft.

3 After each theft, R becomes 4(R−1)/5, due to the removal of one pile.

4 For each trial value of N, Loop 2 checks the value of 4(R−1)/5 *after each* theft.

5 If the remaining number of coconuts after a particular theft is not an integer, increase N by 5 and repeat Loop 1. Set a tentative limit of 20,000 for N.

6 The final share of each sailor, F, should be (R−1)/5 instead of 4(R−1)/5 as given by the 6th iteration in Loop 2. Hence we should have F=R/4.

(2) Backward Algorithm

We can also think of any solution to a problem as a path leading from some given information to a specified goal. Hence, it may be easier sometimes to begin at the specified goal and work our way backwards to unravel the problem at the start. Piele and Wood (1979) solved the "five sailors" problem using such an ingenious strategy.

Before each sailor had received his share of F coconuts in the morning, there were (5F+1) coconuts left over after the fifth theft by the last sailor. This remaining pile was the result of grouping together 4 equal shares with another equal share hidden somewhere by the fifth sailor. Therefore (5F+1)/4 should be some integer K, which represented the number of coconuts stolen by the last sailor from a previous pile of (5K+1) coconuts. Similarly, this pile came from grouping together (5K+1)/4 equal shares by the previous sailor. Again, (5K+1)/4 should be an integer. By "moving backwards" in this manner through all the five sailors, we could formulate a *backward algorithm* (Figure 2) which resulted in a more efficient BASIC program capable of solving the same problem in less than 25 seconds.

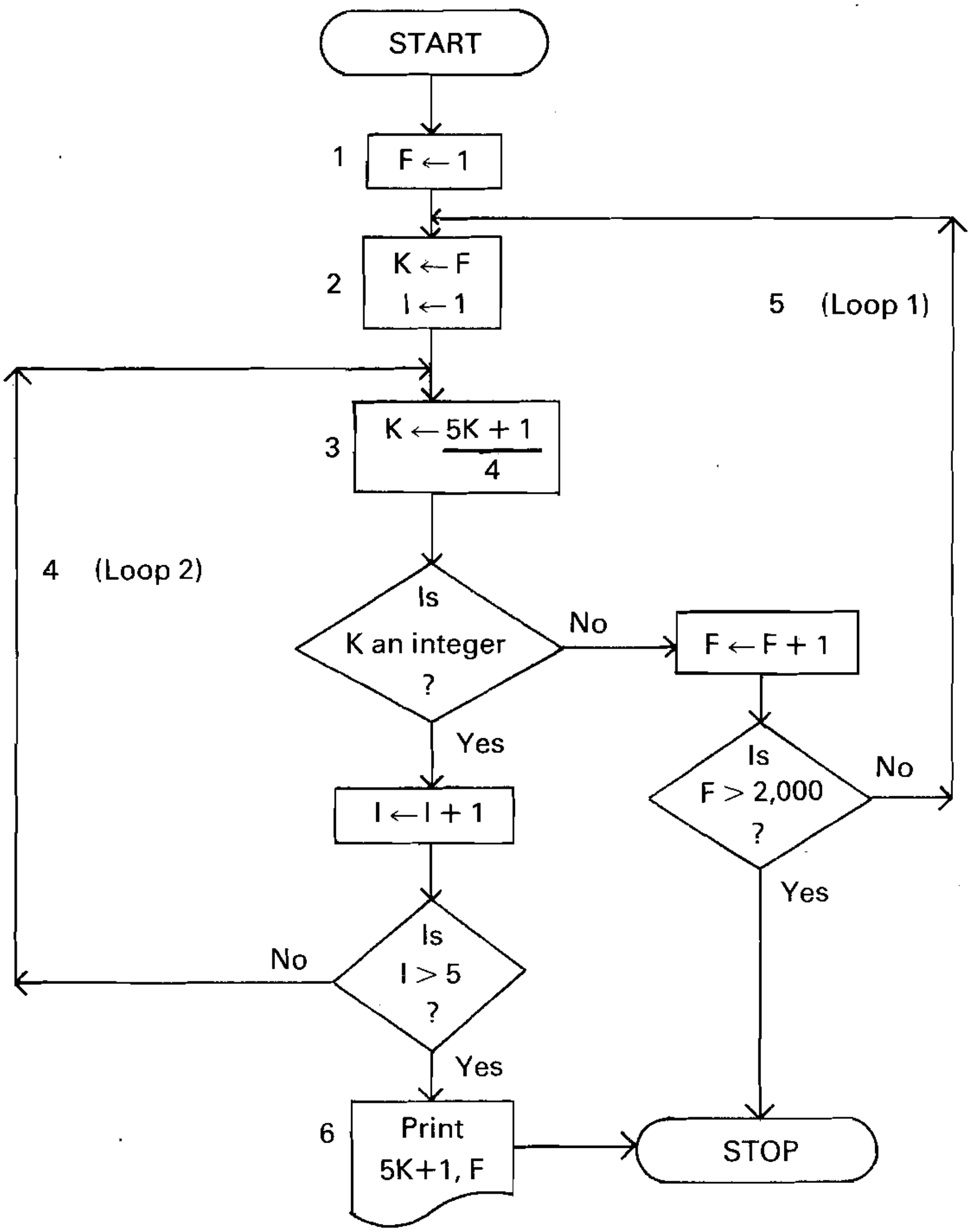

Figure 2. A flowchart for determining the number of coconuts using *backward algorithm.*

Flowchart Notes

1 F=K=1 is the least possible value for the morning share.

2 F is the counter for Loop 1, and I for Loop 2.

3 Let the number of coconuts stolen by a sailor, (5K+1)/4, be the new value of K.

4 For each trial value of F, Loop 2 checks to see whether this final share can survive being pushed back through five consecutive thefts and regrouping and still yield an integer K at each stage.

5 F is increased by 1 in Loop 1 whenever K fails to be an integer. Set a tentative limit of 2,000 for F.

6. (5K+1) gives the least number of coconuts.

```
5   REM —— BACKWARD ALGORITHM ——
10  FOR F = 1 TO 2000
20  K = F
30  FOR I = 1 TO 5
40  K = (5 * K + 1) / 4
50  IF K < > INT (K) THEN 100
60  NEXT I
70  PRINT "LEAST NUMBER OF COCONUTS = "; 5 * K + 1
80  PRINT "EACH SAILOR'S FINAL SHARE = "; F
90  STOP
100 NEXT F
110 END

RUN
LEAST NUMBER OF COCONUTS = 15621
EACH SAILOR'S FINAL SHARE = 1023

BREAK IN 90
```

Conclusion

Instructing students in logical reasoning has always been one of the major goals in mathematics education. The teacher's responsibility to her students is not merely to ensure the acquisition of knowledge in subject areas but also to train students to think and solve problems effectively and efficiently. The conventional thinking in mathematics teaching is to give students more problems to solve. But as students progress through their school years, they soon realize that the complexity of calculations and algebraic manipulations increase exponentially. When classical algebraic or analytical techniques lead to awkward situations which are either too difficult to disentangle or unsolvable with established rules or formulae, students are generally told that more advanced theorems are needed. However, the discussion above has demonstrated the usefulness of algorithms which require only knowledge of elementary mathematics in solving the complicated Diophantine problem. With the introduction of micro-computers into schools, teachers should exploit the high-speed computational capabilities in computers to teach algorithmic strategies of problem-solving.

References

Appel, Kenneth and Haken, Wolfgang. "The Solution of the Four Color-Map Problem", *Scientific American* (Oct. 1977), pp. 108–121.

Brashear, Philip W. "Five Sailors and a Monkey", *Mathematics Teacher* (Oct. 1967), pp. 597–599.

Piele, D.T. and Wood, L.E. "Thinking Strategies With the Computer: Working Backward", *Computers In Mathematics: A Sourcebook of Ideas,* (New Jersey: Creative Computing Press, 1979), pp. 30–32.

Tan Wee Kiat. "Computer-Assisted Solutions To Numerical Puzzles", *Teaching and Learning* (vol. 3, no. 2, Jan 1983), pp. 30–36.