Spring/Fall 2003

Elwyn Berlekamp (berlek@math.berkeley.edu)
Joe P. Buhler (jpb@msri.org)

Note: These solutions are a work in progress; comments, references, etc. are appreciated. Most references and figures can be found with the problem statements.

Problem 1. One hundred ants are placed on a stick one meter long. Each ant begins to travel either to the left or to the right at a constant speed of one meter per minute. When two ants meet, they bounce off and reverse direction while maintaining their speed. When an ant reaches either end of the stick it falls off.

What is the longest amount of time one must wait to be sure that the stick is completely ant-free?

Discussion: This is the first of several problems about ants. In every case, it is very useful to perform the following thought experiment: every ant carries a baton; whenever a pair of ants bounce off of each other they exchange their batons. Thus every ant always has one baton, and the directions and velocities of the batons are unaffected by collisions between the ants.

This thought experiment is sufficient to solve this problem: all batons move forward at the rate of one meter per minute. After one minute all batons have fallen off of the stick, and all of the ants have gone with them. One minute is also the least possible upper bound on the time: a baton starting very close to one end and traveling towards the other end will take nearly one minute to traverse the full length of the stick and fall off.

Problem 2. Let n be a positive integer. (a) Show that some multiple of n has only 0's and 1's in its ordinary base 10 representation. (b) Show that some multiple of 2^n has only 1's and 2's in its base 10 representation.

Discussion: The solution to part (a) in Winkler's book is particularly elegant (and avoids special cases): Given n, consider the integers $(10^k - 1)/9$ whose decimal expansions consist of a string of k consecutive 1's. Some two of them are congruent modulo n and their difference is a multiple of n of the desired form.

In the case of (b) a stronger assertion can be proved by induction on n: there is an integer with exactly n digits,in its decimal expansion, all 1's or 2's, that is divisible by 2^n. Indeed, if x_n is the decimal expansion of such an n-digit integer, then exactly one of $1x_n$ or $2x_n$ is an $n+1$-digit integer divisible by 2^{n+1}.

Problem 3. How many disks of diameter 1 fit in a $2 \times N$ rectangle?

Discussion: The obvious Cartesian packing places $2N$ disks of diameter one into a $Y \times N$ rectangle when $Y = 2$. But we can do better if N is sufficiently large. For motivation, it helps to imagine that Y is very slightly less than 2. Then we might stagger the top and bottom rows, as follows:

$$\begin{array}{cccccccccccc} & B & & D & & F & & B & & D & & F & \ldots \\ A & & C & & E & & A & & C & & E & & \ldots \end{array}$$

This works especially well if the height $Y = 1 + \sqrt{3}/2 = 1.8660\ldots$. Except for the ends, each disk touches 1 wall and 4 other disks, so the "kissing number" is 5. Measured from the west wall, the longitudes of the centers are at 0.5, 1.0, 1.5, 2.0, 2.5, ... If there are N disks, the total length is $1 + N/2$.

Now if the height is increased slightly to some value of Y slightly greater than $1.8660\ldots$, it is wise to retain essentially the same packing pattern, except that we should now be careful that each of the equilateral BCD triangles touches the top edge, whereas each of the EFA triangles should touch the bottom edge. We can then reduce X by squeezing everything westward. The kissing number is then 4, because A no longer touches B; D no longer touches E; and C and F no longer touch any wall. The centers of E and A are still at elevation $1/2$, but the elevation of the center of C is now $1/2 + 1 - \sqrt{3}/2$. So if x denotes the longitudinal separation between the centers of A and C, then, according to Pythagoras,

$$x^2 + (1 - \sqrt{3}/2)^2 = 1$$

whence

$$x = \sqrt{\sqrt{3} - 3/4)} = 1 - \delta$$

where $\delta = 1/110.923362\ldots$.

The length of the smallest $2 \times L$ rectangle holding the N coins arranged as $ABCDEFABCDEF\ldots$ as described above is

$$L(N) = (N+1)/2 - \lfloor \frac{n+1}{3} \rfloor \cdot \delta.$$

This formula can be readily verified for each of the cases $N = 1, 2, \ldots, 6$. Results for larger values of N follow from the periodicity of the packing pattern.

Comparing the Cartesian and staggered packings of a $2 \times \lceil X \rceil$ rectangle, we find that: Cartesian wins if $\lceil X) \rceil \leq 83$, they tie if $84 \leq \lceil X \rceil \leq 165$. Staggering wins with one more disk if $166 \leq \lceil X \rceil \leq 248$, and staggering wins with two more disks if $249 \leq \lceil X \leq 330$

Here is a possible generalization: How many nonoverlapping coins of diameter 1 can be packed into a $Y \times X$ rectangle, when $X \gg Y$?

Here is a sketch of a conjectured solution.

Begin by defining $R = 1 + (Y-1)2/\sqrt{3}$.

If R is an integer, then hexagonally pack with R rows. Else partition the coins into sets of $\lfloor R/2 \rfloor \cdot \lceil R/2 \rceil$ coins. Pack the coins in each set into an equilateral triangle. Let half of such triangles touch the bottom of the rectangle, and alternate between these and the other (inverted) half of the triangles, which touch the top. Fill in the gaps along the western and eastern edges of the rectangle with about half of a triangle.

Problem 4. A chess game ends on White's sixth move, which is g×f8N, checkmate. (This means that a pawn on KN7 captures a piece on KB8 — the home square of the Black king's bishop — promoting to a knight and checkmating.) Reconstruct the game.

Discussion: The (rather unlikely) game is: **1.** P-KR4, P-Q4 ; **2.** P-KR5, N-Q2 ; **3.** P-KR6, QN-KB3 ; **4.** P×P, K-Q2 ; **5.** R-KR6, N-Q1 ; **6.** P×B. (i.e., 1.h4 d5 2.h5 Nd7 3.h6 Ndf6 4.h×g7 Kd7 5.Rh6 Ne8 6.g×f8N mate). As mentioned in the Elkies and Stanley article, this remarkable construction is due to P. Rössler.

Problem 5. We have a circular key chain and want to color the keys, using as few colors as possible, so that each key can be identified by the color pattern — that is, by looking at they key's color and neighboring colors as far away as needed. Let $f(n)$ be the minimal number of colors required to uniquely disambiguate a circular key chain of n keys in this way. Determine $f(n)$ for all positive integers n.

For instance, for $n = 4$ three colors suffice: the keys can be identified as green, blue, red next to green, red next to blue. By checking cases one sees that two colors is impossible (key rings can be flipped over, so references to left, right, clockwise etc. are forbidden). All in all, we conclude that $f(4) = 3$.

Discussion: With n given, we want to find the minimum $k = f(n)$ such that there is a chain C of length n chosen from $1, 2, \ldots, k$ such that any cyclic rotation of C, or reflection and cyclic rotation, is distinct from C. For $n = 2, 3, 4$, the chains 12, 123, and 1123 are easily checked to be optimal. For $n \geq 5$, the chain 2122111 . . . works. Indeed, any reflection will have two 2's to the left of the isolated 1, and any cyclic rotation has the isolated 1 in a different position. Thus $f(n) = 2$ for all $n \geq 5$.

Problem 6. A read-only array of length n contains entries $a[0], a[1], \ldots, a[n-1]$ all taken from the set $\{1, \ldots, n-1\}$. By the pigeonhole principle there is a duplicated element (at least one).

(a) Assume that there is exactly one duplicated entry. Find an algorithm (e.g., a program in your favorite programming language) that takes this array as input and then prints out the duplicated value. The program cannot modify the input array. It should run in linear time, and it should use a constant amount of storage words (each capable of storing integers up to n).

(b) Write a program, subject to the same constraints, that prints a duplicated element, without making any assumptions on the number of duplicated entries. This program, too, is forbidden from modifying the input, should run in linear time, and should use constant extra space.

Discussion: (a) The duplicated integer is easy to figure out by comparing the sum of all of the elements of the array with the sum of all integers between 1 and $n-1$.

Even more explicitly, the duplicated integer is equal to the exclusive-or of all array entries and all integers between 1 and $n-1$.

(Bonus problem that speeds up this algorithm by allowing the ex-or of 1 though $n-1$ to be precomputed quickly: show that the ex-or of 1 through $4k+3$ is 0, for any nonnegative k.

And now for part (b). The array a defines a (perhaps "random") function from the set $\{0, \ldots, n-1\}$ to itself via $f(i) = a[i]$. Since the values are positive, the function never takes the value 0 and it follows that the sequence

$$0,\ f(0),\ f(f(0)),\ f(f(f(0))), \cdots$$

must repeat at a nonzero value. I.e., if we let $f^{(k)}$ denote the k-fold composition of f, then there is are positive r and s such that

$$f^{(r)}(0) = f^{(s)}(0), \qquad r < s.$$

If r and s are the smallest such values, then $x := f^{(r-1)}(0)$ and $y := f^{(s-1)}$ are distinct, and $a[x] = a[y]$.

This idea is familiar in several other contexts (e.g., Pollard's ρ factorization method), and there is a cute algorithm that finds a duplicated value:

$i := a[0]$
$j := a[i]$
while i is not equal to j:
    $i := a[i], \quad j := a[a[j]]$
$j = 0$;
while i is not equal to j :
    $i := x[i],\ j := x[j]$

The reader will have to check several things to see that i is a duplicated value at the end of the second loop. E.g., since i is stepped once and j is stepped twice in the first loop, sooner or later they will be equal; moreover, since i and j start a distance i apart at the beginning of the second loop, sooner or later they are equal, and they are a duplicated value the first time that occurs.