The Four-Cubes Four-Color Problem

or Instant Insanity

by

Larry Axsom and Peter Wegner

Technical Report

No. 69-30

January 1969

Department of Computer Science
Cornell University
Ithaca, New York 14850

The Four-Cubes Four-Color Problem
or Instant Insanity

by

Larry Axsom and Peter Wegner

1. The Representation of Cubes

A puzzle widely marketed under the name "Instant Insanity" involves four cubes each having their six faces colored with a combination of the colors red, green, white and blue. The object of the puzzle is to arrange the four cubes in sequence so that each set of four faces pointing in a given direction has four differently colored faces.

This paper presents a method of generating all solutions to the puzzle by enumerating all possible different arrangements of cubes, considers how the number of enumerations can be reduced by excluding classes of arrangements which cannot contain solutions and by taking the symmetry of solutions into account, and discusses the implementation of the method and the results obtained.

The method described below will generate all solutions for an arbitrary coloring of faces of the cubes. However, we shall illustrate the method by considering the set of face colorings of the puzzle that is marketed.

The first problem in developing the enumeration technique is to choose a representation of cubes that allows the enumeration of arrangements to be easily performed. In order to develop such a representation we shall first characterize the faces of the cube by the direction in which they point as illustrated in Figure 1.

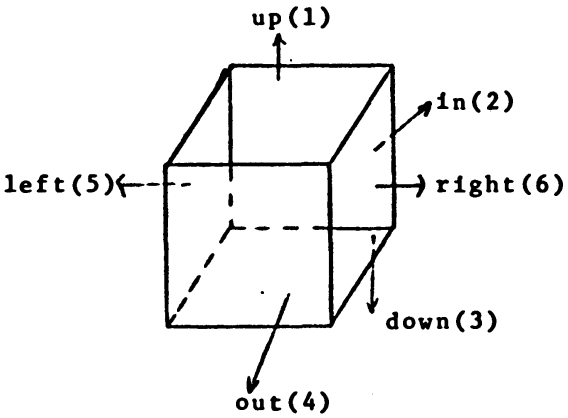

Figure 1. Names and Indexes of Cube Faces

In Figure 1, the six faces are called the up-face, in-face, down-face, out-face, left-face and right-face, and are associated with the respective indexes 1, 2, 3, 4, 5, 6. Figure 2 illustrates a two-dimensional representation of the six faces, and specifies a coloring of one of the cubes of the marketed puzzle in this representation.

	up(1)			white	
	in(2)			red	
left(5)	down(3)	right(6)	red	blue	green
	out(4)			red	

Figure 2. a) Two-dimensional Representation
b) Cube Coloring

Figure 3 specifies an arrangement of the four cubes of the marketed puzzle that is not a solution.

	white			white			white			white	
	red			green			white			blue	
red	blue	green	red	blue	blue	red	green	green	red	green	white
	red			green			blue			red	
	CUBE 1.			CUBE 2.			CUBE 3.			CUBE 4.	

Figure 3. An Arrangement of the Four Cubes.

It is evident from Figure 3 that the four cubes have inherently different coloring patterns in the sense that there is no change of orientation of any cube which can convert it to the same coloring pattern as one of the other cubes. This follows from the fact that no two cubes have the same number of faces of each color.

Since solutions depend only on the orientation of each cube and not on the order in which cubes are arranged, we shall consider only arrangements in which cubes appear in the left to right order of Figure 3. The four cubes will be named Cube 1, Cube 2, Cube 3 and Cube 4 as in Figure 3.

Figure 4 illustrates an arrangement of the four cubes that is a solution.

	red			white			green			blue	
	blue			red			green			white	
red	green	red	green	blue	green	blue	white	white	white	red	red
	white			blue			red			green	
	CUBE 1.			CUBE 2.			CUBE 3.			CUBE 4.	

Figure 4. An Arrangement of the Cubes of Figure 3 which is a Solution.

The fact that Figure 4 represents a solution follows from the fact that the four rows obtained from corresponding elements of the central column of each cube have different colors. It is not, however, immediately obvious that each of the cubes of Figure 4 can be obtained by a reorientation of the corresponding cube of Figure 3. The reader who wishes to gain some insight into the problem of representing cubes might wish to characterize the reorientation required to transform the cubes of Figure 3 to those of Figure 4 before reading how the authors have solved the problem of characterizing reorientation.

It is convenient for computational purposes to represent an orientation of a single cube by a six-element vector, and an arrangement of the four cubes by a six by four matrix whose

columns represent the orientation of individual cubes. Using the ordering of faces indicated by the indexes of Figure 1, the arrangement of Figure 3 is represented by the following six by four matrix.

white	white	white	white
red	green	white	blue
blue	blue	green	green
red	green	blue	red
red	red	red	red
green	blue	green	white

Figure 5. Matrix Representation of Figure 3.

For a given orientation of a cube the left and right faces will be called <u>hidden faces</u>, and the up, in, down and out faces will be called <u>exposed faces</u>. In the matrix representation of the cube the exposed faces of the cube are represented by rows 1, 2, 3 and 4, and the hidden faces are represented by rows 5 and 6. A given arrangement of the four cubes is a solution if, in the matrix representation, each of the first four rows contains four different colors.

The number of different orientations of a single cube is 24. i.e. There are six ways of identifying a cube face with a given direction (say the left direction), and for each of these there are four different orientations of the remaining faces obtained by rotation about an axis perpendicular to the left direction.

The number of different arrangements of a sequence of four cubes is $24^4 \sim 330{,}000$, since each of the cubes can independently be oriented in 24 ways.

Note that there are 24 ways of listing the cubes in sequence. The decision to only consider the set of solutions associated with one of these 24 orderings cuts down the enumeration by a factor of 24. This decision is justified by the fact that knowledge of the set of all solutions for one ordering of cubes allows the set of all solutions for any other ordering to be trivially generated. The general conditions under which this reasoning can be applied to reduce the number of enumerated cases is indicated in the footnote*.

---

* The reasoning which justifies restricting the enumeration to one ordering of cubes can be formulated mathematically in terms of equivalence and congruence relations. Let a be an arrangement of cubes and let ϕ be a class of mappings which reorders cubes wihtout disturbing their orientation. Then any arrangement $a' = \phi_i(a)$ obtained by applying $\phi_i \in \phi$ to an arrangement a, is a solution if and only if a is a solution. i.e. Mappings $\phi_i \in \phi$ preserve the property of being a solution.

Let two arrangements be defined to be equivalent if one can be obtained from the other by a mapping in the class ϕ. Let $\bar{a}$ be the equivalence class of all arrangements equivalent to a under the mapping ϕ, and let Φ be the equivalence relation over arrangements a determined by the equivalence classes $\bar{a}$. Let s be a function which, when applied to an arrangement a yields the value true if a is a solution and false otherwise. The equivalence relation Φ is said to be a congruence relation for the function s since the result $s(a)$ of applying s to an arrangement a depends only on the equivalence class $\bar{a}$ of a and not on the individual element a.

The fact that Φ is a congruence relation over arrangements a for the function s allows us to develop an algorithm which systematically selects precisely one element of each equivalence class, and reduces the number of enumerated items by avoiding multiple enumeration of elements in a given equivalence class. The element of the equivalence class chosen by the enumeration is sometimes referred to as the representative element or normal form for elements in the equivalence class.

*(contd.) The general condition for this approach to be applicable is as follows:

If an equivalence relation $\overline{\phi}$ over arrangements a can be found which is a congruence relation for the solution function s , and if a procedure for systematically enumerating precisely one element of each equivalence class $\overline{a}$ can be found, then this procedure can be used to reduce the number of enumerated items to one item per equivalence class $\overline{a}$. The representative element of $\overline{a}$ will determine whether elements of $\overline{a}$ are solutions. If enumeration of one of the equivalence classes $\overline{a}$ is desired this can be done by an independent enumeration procedure.

2, The Enumeration Technique

There are many alternative orders in which the 24^4 arrangements of cubes could be enumerated. For example, we could run through all possible orientations of the first cube for fixed orientations of the second, third and fourth cubes, then consider all orientations of the first two cubes for fixed orientations of the third and fourth cubes, all orientations of the first three cubes for a fixed orientation of the fourth cube and finally all orientations of the first four cubes. This approach corresponds to thinking of each arrangement as a four digit number in the base 24 and enumerating successive arrangements by counting in the base 24.

The approach to enumeration adopted here is more highly structured than the above approach, and was chosen because it lends itself more readily to the elimination of unproductive alternatives.

The 24 orientations of a given cube are characterized by the following three-way classification

1. For each cube there are three ways of choosing pairs of hidden faces. A fixed choice of hidden faces will be called a hidden-face configuration, and denoted by an integer $d_i = 0,1$ or 2.

2. For each hidden-face configuration there are two ways of orienting the pair of hidden faces. An orientation of hidden faces within a given hidden-face configuration will be called a hidden-face orientation, and denoted by an integer $e_i = 0$ or 1 .

3. For a given hidden-face orientation, the cube can be rotated through four positions. Each rotated position within a given face orientation is called a rotation, and denoted by an integer $f_i = 0,1,2$ or 3 .

A given orientation of a cube i is thus characterized by a triple (d_i,e_i,f_i) where d_i can take three different values, e_i can take two different values and f_i can take four different values. Thus the triple (d_i,e_i,f_i) can take 24 different values each characterizing a different orientation.

The hidden-face configuration of all four cubes is characterized by a four digit vector $d = (d_1d_2d_3d_4)$. There is a total of $3^4 = 81$ hidden-face configurations.

The hidden-face orientation of all four cubes for a fixed hidden face configuration is characterized by a four-digit vector $e = (e_1e_2e_3e_4)$. There are $2^4 = 16$ hidden-face orientations for a given hidden-face configuration.

The rotation of all four cubes for a fixed hidden-face orientation is characterized by a four-digit vector $f = (f_1f_2f_3f_4)$. There are $4^4 = 256$ rotations for a given hidden-face orientation.

The enumeration technique described below enumerates all rotations for a fixed hidden-face orientation, then runs through all hidden-face orientations for a given hidden-face configuration, and finally runs through all hidden-face configurations. Each hidden-face configuration requires $2^4 \times 4^4 = 4096$ cases to be enumerated.

The above enumeration can be thought of as a counting procedure on a twelve digit number $(d,e,f) = (d_4 d_3 d_2 d_1 e_4 e_3 e_2 e_1 f_4 f_3 f_2 f_1)$ where counting for digits f proceeds modulo 4, counting for digits e proceeds modulo 2 and counting for digits d proceeds modulo 3. The total number of states of this 12-digit number is $3^4 \times 2^4 \times 4^4 = 24^4$. However, the 24^4 arrangements are enumerated in a different order from that which would have been obtained by running through all orientations of successive cubes.

Having chosen the order of enumeration, it is necessary to consider how the transition from one arrangement to the next is to be accomplished. An arrangement is specified by a six by four matrix, and it would in principle be possible to specify the transition from one arrangement to the next by a mapping of the matrix into itself. However, different mapping functions must be applied at different points of the enumeration depending on carries and overflow from one digit position to the next in the twelve-digit number representing the current arrangement. Complex checks would have to be devised to ensure that the correct mapping was always performed. A more uniform mapping procedure which determines the matrix directly from the twelve-digit number is described below.

The enumeration procedure starts with an initial arrangement, say that in Figure 3, and $d = e = f = (0,0,0,0)$. The matrix associated with the initial arrangement will be denoted by M and referred to as the reference matrix. The reference matrix never changes throughout the enumeration.

The transition from one arrangement to the next will be accomplished with the aid of three additional matrices D , E , F , where D represents the current hidden-face configuration, E represents the current hidden-face orientation for the given hidden-face configuration, and F represents the current rotation for the given hidden-face orientation. Initially, the matrices D , E and F are identical to the references matrix M . The updating of the matrices D , E , F during the enumeration of arrangements is described below.

Every cube has an initial hidden-face configuration specified by the reference matrix M , and two additional hidden face configurations. The two additional hidden-face configurations can be obtained by permuting the faces of the cube. A permutation of the faces of the cube may in turn be represented by a corresponding permutation of the six elements of the column vector representing the cube in the matrix M .

The initial hidden-face configuration which is represented by the value $d_1 = 0$ is associated with the identity permutation (123456) . Let $d_1 = 1$ be associated with a ninety degree anticlockwise rotation about an axis pointing in the "in" direction perpendicular to faces 2 and 4, and let $d_i = 2$ be associated with a ninety degree anticlockwise rotation about an axis in the down direction perpendicular to faces 1 and 3. The permutation vectors associated with the above rotations are, respectively, (625413) and (163524) .

The matrix D for a given initial face configuration $d = (d_1 d_2 d_3 d_4)$ is obtained by mapping column i of M using the permutation vector associated with the digit d_i . The resulting matrix D is said to be an initial hidden-face configuration, and is used as a reference matrix for computing the 16 matrices E which represent hidden-face orientations for a given hidden-face configuration.

The mapping of the ith column of D to the ith column of E is given by the identity permutation (123456) if $e_i = 0$. The orientation $e_i = 1$ represents an interchange of the faces 5 and 6. If this is done by a rotation of 180 degrees about an axis perpendicular to faces 1 and 3, the corresponding column permutation is (143265) . The matrix E is recomputed from the matrix D whenever a new hidden-face orientation is required by permuting columns of D as specified by $(e_1 e_2 e_3 e_4)$.

The matrix F represents a rotation of the matrix E , and is obtained from E by applying a rotation permutation to the ith column specified by the digit f_i . The rotation

permutations associated with the digits $f_i = 0,1,2,3$ are respectively (123456) (234156) (341256) and (412356) .

In the initial version of the program, the complete mapping from E to F was performed for every arrangement, the mapping from D to E was performed whenever f overflowed and e was incremented, and the mapping from M to D was performed whenever e overflowed and d was incremented.

In later versions of the program, the mapping from E to F , which is in the inner loop was replaced by direct mapping from one rotation to the next.

The flow diagram for the complete enumeration of the 24^4 arrangements is as follows:

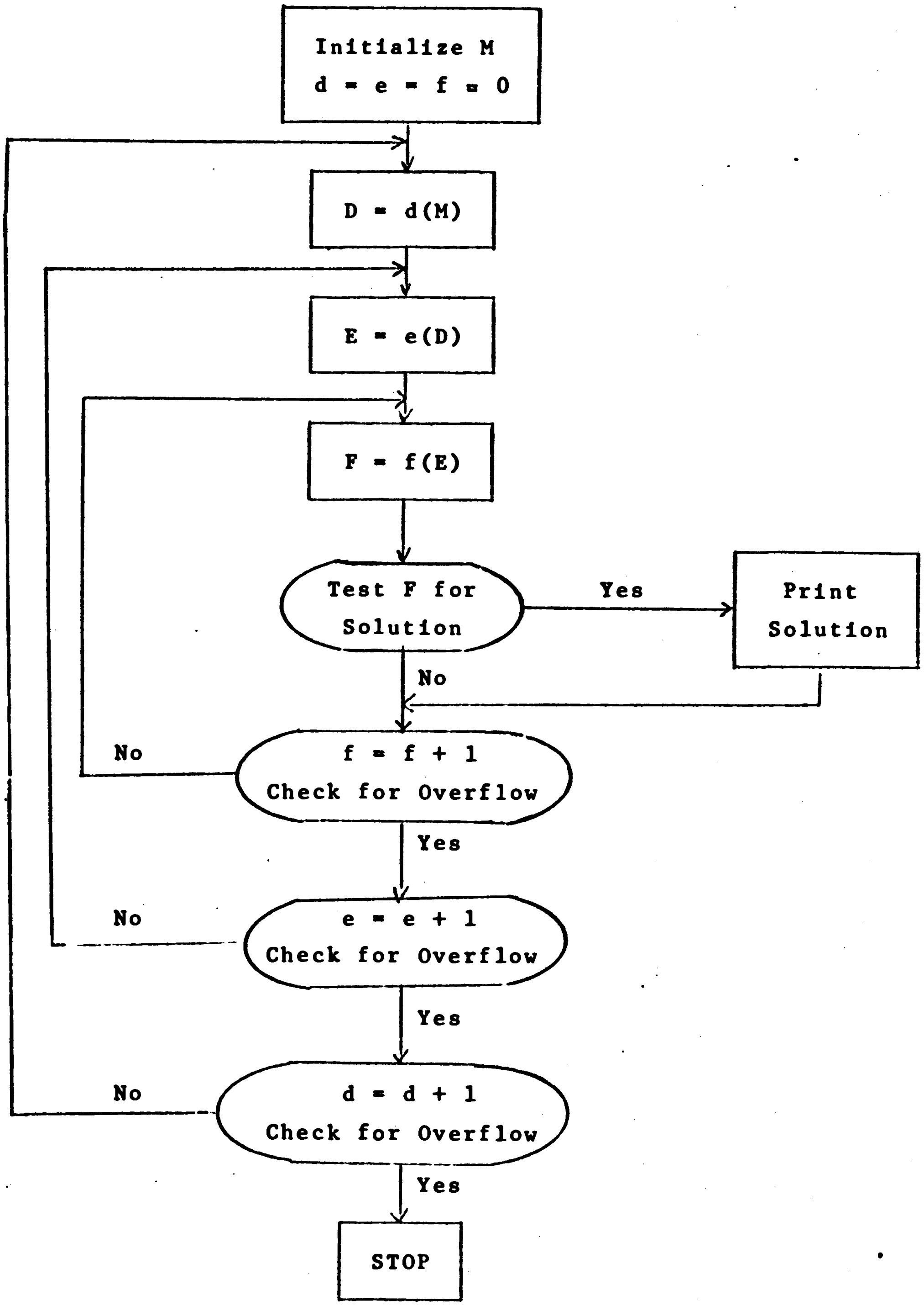

Figure 6. Flow Diagram for Complete Enumeration

In order to gain some insight into the nature of the permutations associated with mappings d , e and f , the orientation (d_1,e_1,f_1) of Cube 1 in Figure 4 relative to Cube 1 in Figure 3 will be determined.

The hidden face configuration of Cube 1 in Figure 4 requires faces 2 and 4 of Figure 3 to be rotated into faces 5 and 6, corresponding to $d_1 = 2$. The permutation (163524) produces the following initial hidden-face configuration

	white	
	green	
red	blue	red
	red	

The configuration of Cube 1 in Figure 4 cannot be obtained from the above configurations by simple rotation, so that the permutation (143265) associated with $e_1 = 1$ must be applied, yielding the following hidden-face orientation

	white	
	red	
red	blue	red
	green	

The orientation of Figure 3 can now be obtained by applying the rotation (234156) , corresponding to $f_1 = 1$. The orientation of Cube 1 in Figure 4 is thus characterized by the coordinates (2,1,1) relative to the orientation of Cube 1 in Figure 3.

The reader may determine the coordinates (d_2,e_2,f_2), (d_3,e_3,f_3), (d_4,e_4,f_4) for Cubes 2, 3 and 4 by a similar technique.

3. Pruning Algorithms

There are two principal classes of algorithms for pruning umproductive enumerations:

1. Certain configurations are equivalent to others in the sense that they can be obtained from a given standard configuration (normal form) by a simple mapping which does not change the color configuration of exposed faces. The pruning of multiple orderings was an example of such an algorithm and was discussed at the end of section one. Other examples of this kind of algorithm are discussed below.

2. Sometimes, necessary (but not sufficient) conditions for the existence of a solution can be found, such that classes of arrangements exist which do not satisfy this necessary condition and need not be enumerated. Two examples of such necessary conditions are discussed below.

We shall first consider two algorithms of the first type for eliminating "symmetric arrangements", and then consider algorithms of the second type.

Each arrangement has three equivalent arrangements that can be obtained from the given arrangement by rotating the complete arrangement about an axis perpendicular to the hidden faces. Generation of such equivalent arrangements can be avoided by permitting only three cubes to rotate and requiring

the fourth to remain fixed during the set of rotations associated with a given hidden-face orientation. This cuts the total enumerations by a factor of four.

Fixing of a given cube is equivalent to fixing the corresponding component f_1 of f. This can be simply implemented by reducing f to a three-element vector $f_1 f_2 f_3$ and moving to a new hidden-face orientation when $f_1 f_2 f_3$ exceeds capacity.

Each hidden-face orientation has an equivalent orientation obtained by rotating all cubes simultaneously about an axis perpendicular to faces 2 and 4 (or 1 and 3). Generation of this equivalent orientation can be avoided by fixing the hidden face orientation of one of the cubes. This reduces the set of enumerated arrangements by a further factor of 2. It can be implemented by reducing e to a three-element vector $(e_1 e_2 e_3)$ and moving to a new hidden-face configuration whenever e exceeds capacity.

The above simple modifications of the enumeration procedure reduce the number of enumerated arrangements by a factor of eight from over three hundred thousand to about forty thousand. We shall now consider ways of further reducing the number of enumerated arrangements by excluding groups of arrangements that can never generate a solution because certain necessary conditions for generating a solution are not satisfied.

Examination of the four cubes of Figure 3 shows that there is a total of seven red faces, six green faces, six white faces and five blue faces. A necessary condition for the existence of a solution is that there be four exposed faces of each color. Therefore any solution must have three hidden red faces, two hidden green faces, two hidden white faces, and one hidden blue face. Hidden face configurations not having this property need not be enumerated, resulting in the elimination of $4^3 \times 2^3 = 512$ arrangements for every eliminated hidden face configuration.

A check on the colorings of hidden faces can be inserted into the flow diagram immediately following the operation $D = d(M)$. If the check indicates an incompatible set of hidden colors, control can be transferred to the point in the flow diagram at which the operation $d = d + 1$ is performed.

In the case of the marketed puzzle, only five of the eighty-one hidden face configurations are color compatible, so that this check reduces the number of enumerated arrangements from forty thousand to $5 \times 512 = 2560$.

The above reductions make the problem into a tractable one which can be solved in a short time on a computer. However, enumeration can be further reduced by applying a further necessary criterion for solutions to exist.

Incompatability of the exposed faces of two cubes or of three cubes implies that no rotation of the remaining cubes

is compatible with a solution. The rotation procedure for a given hidden face orientation can be modified so that the compatibility between cubes 3 and 4 is checked both initially and whenever f_3 is incremented. Incompatiability of cubes 3 and 4 gives rise to immediate incrementation of f_3 without enumerating rotation of cubes 1 and 2. Similarly a check for compatability of cubes 2, 3 and 4 can be introduced whenever f_2 is incremented and rotations of cube 1 can be skipped for all incompatible colorings of cubes 2, 3 and 4. These checks further reduce the number of arrangements actually enumerated.

It should be noted that all of the above techniques for systematically reducing the number of enumerated arrangements correspond to techniques that the human problem solver would automatically use in solving the problem. The present solution procedure is therefore a good illustration of the formalization of techniques that correspond to "intelligent" behavior.

The human problem solver is better at applying reduction algorithms which exclude enumeration of instances not satisfying necessary conditions than at applying reduction algorithms which take symmetry into account. In the present case the advantage of the computer lies in being able to simultaneously apply several reduction algorithms which are individually sufficiently simple for the user to apply, but which are collectively too complex for the user.

The brand of insanity typified by this example is clearly that which results from the combination against the user of forces with which he knows he could deal individually but which collectively overwhelm him.

4. Implementation

The above project was conceived at about 7 pm one evening when the authors were visiting the house of a friend X in which failure to solve the problem was causing great domestic strife. In response to an appeal to solve the problem, the authors started formulating a solution at 8 pm over dinner. By 10 pm the outlines of an algorithm had emerged and it was decided to spend the night programming the solution on a computer with an APL system. A call was put through to X indicating that the blocks were needed that evening to provide data for our algorithm. It was agreed that he would leave them on his front porch so that we could pick them up later that evening.

When we arrived at X's house at 11 pm, the blocks were not there and we rang the bell assuming that he was still up. Mrs. X came down in her night robe, and we apologized profusely and asked her why the blocks had not been put out. She said that she had indeed noticed the blocks on the front porch about 15 minutes ago, but had interpreted this as a gesture of extreme exasperation on the part of her husband at not being able to solve the problem and had taken them back in.

On getting the blocks the authors went to the computation center at about midnight and started programming. By 2 am a program for generating the hidden-face configurations and checking for color compatibility had been successfully debugged and it was determined that only five of the eighty-one solution were color compatible. While one of the authors continued programming at the APL console, the other author tried to generate solutions manually from the five hidden face configurations printed by the computer and found a solution while trying the third of the five hidden-face configurations. This solution was later used to debug certain paths in the program, providing the authors with one of the first authentic illustrations of the value of interaction between humans and computers.

The first working algorithm was produced at about 5 am. It included a check for hidden face compatibility but no elimination of rotational symmetries, and took about two hours to run.

The long running time provided time for reflection, and an algorithm which took account of rotational symmetries and the color compatability of partial arrangements was developed. By 8 am an algorithm with a running time of 1 1/2 minnutes had been debugged.

It was proved that there was only one solution to the problem* and this solution was triumphantly presented to Mr. and Mrs. X.

* There is only one solution when the $24\times8=192$ symmetrical solutions are regarded as equivalent. The 192 symmetrical solution can be trivially generated from the single computer-generated solution (see footnote at the end of section 1).