What is Discrete Mathematics and how should we teach it?

In the past 25 years the rôle of *discrete mathematics* has become increasingly important. The number of fields in which discrete mathematics is applied in some way also keeps increasing. It has been argued that for some areas, where mathematical knowledge is necessary, one should *replace* the standard calculus course by discrete mathematics. Although I feel that everybody should know some calculus, it is certainly true that knowledge of techniques from discrete mathematics is often just as useful. A number of years ago this idea of replacing calculus by parts of mathematics that were more relevant to the other part of the program was pushed strongly by computer science departments in the USA. This led to a stream of books on "Discrete Mathematics for Computer Scientists", most of which give the impression that discrete mathematics is the union of all subjects in mathematics that are useful for computer scientists but not part of calculus. One finds logic and set theory as part of the hodge podge of subjects in these books. My opinion is that this is *not* discrete mathematics at all! Of course logic, set theory, etc. are very useful for students of computer science but a course in these subjects should be given some other name.

What is discrete mathematics? It is that part of mathematics that deals with with discrete structures. Usually the objects that are studied are finite but of course I also include infinite graphs, the integers, (and other locally finite structures). Essentially the subject includes combinatorial theory, elementary number theory, finite groups, finite geometries, finite fields, and some newer areas such as coding theory.

It is my impression that many courses that do deserve the name discrete mathematics are taught in a way that leaves students completely baffled. They have the impression that problems in discrete mathematics are solved by ingenious tricks and that any new problem that they will encounter requires them to invent the appropriate new trick. Compare this to a calculus course, where one teaches *methods* such as differentiation, integration, solving linear differential equations, etc. and subsequently applies these methods in several *different* situations. A course in discrete mathematics should be similar! One should treat *objects* that appear in many places, sometimes disguised; methods of *representation* should be used in several different situations; *ideas* that reappear regularly in practice should also reappear regularly in the course; *tools* that play an important rôle in discrete mathematics should become part of the skills of the students. To give an idea I mention several examples of each of these topics (not a complete list) :

(1) Objects : graphs, lattices, geometries, designs, codes, coverings, partitions, systems of sets, matroids;
(2) Representations : addressing schemes, coding, (0,1)-matrices, (0,1)-sequences, graphs, diagrams, pictures, subsets of lattices;
(3) Ideas : counting techniques, probabilistic techniques, (non-)existence methods, construction techniques, unification (association schemes, matroids), optimization methods, max-flow, search techniques, symmetry;
(4) Tools : algebra (matrix theory, finite groups, finite fields, group rings), elementary number theory, permutation groups, geometry, analysis (power series, Lagrange inversion).

The course should be structured as a multipartite graph with subsets of (1) to (4) as independent sets and as many edges as possible. Here an edge from say "graph"to "(0,1)-matrices"means that this representation is used to describe graphs but also also used to derive properties of graphs or to prove theorems about them.

The following situation can (should) occur. It has didactical value. One wishes to prove a certain theorem about, say, designs and decides to use (0,1)-matrices as representation. The rows of the (0,1)-matrix can also be interpreted as words in a code. This leads to a formulation of the theorem, that is to be proved, in another terminology. This other theorem may have already occurred in the course or it could be much easier to see how to prove it. One can also prove a "new"theorem about some combinatorial object and in retrospect observe that if this object had been represented in the appropriate way, one would have realized that the theorem had actually occurred earlier in some other form.

If the instructor decides to take the tool "algebra"as a central item in his course, then the ideas that he uses, for example eigenvalues of matrices, should be applied for many different purposes, such as nonexistence theorems for strongly regular graphs, properties of block designs, theorems in finite geometry. Similarly, the idea of using several small combinatorial objects to construct one large one should reappear (Latin squares, Hadamard matrices, block designs, etc.).

A course taught in Eindhoven for several years started with a chapter on finite fields and then chose a number of objects from combinatorics (Latin squares, Hadamard matrices, finite geometries, block designs, error-correcting codes) in *each* of which finite fields were heavily used to construct the objects.

A number of ideas that I use will be treated below as examples. First however, I mention a principle that was suggested by A. Revuz at the meeting on "How to teach mathematics so as to be useful" held in Utrecht in 1967. I have used it ever since with much success. Discrete mathematics is particularly suited for this principle. The idea is to let the students work on problems (usually in groups of two or three), solutions to be handed in as homework, and to teach the standard techniques and theorems necessary to solve the problems a few weeks *later*! Usually one sees several students in class recognize how useful a theorem is long before the proof is finished; (if I had known that idea two weeks ago, then ...).

The use of representations.

If possible, use representations of combinatorial objects not only as representations but in such a way that the chosen representation makes it easier to prove the theorem in question.

EXAMPLE 1. A puzzle known as *Instant Insanity*, involving stacking up multicolored cubes in some way (treated in many books on graphs), is extremely difficult, as the name suggests. It becomes practically trivial when the cubes are represented by graphs that reflect the color-structure.

EXAMPLE 2. A well known way of representing a partition is a so-called Ferrers diagram. Such a diagram actually is a representation of two partitions. This makes it possible to prove theorems of the type "The number of partitions of an integer with property I equals the number of partitions with property II" by just looking at the diagrams.

EXAMPLE 3. Binary rooted trees can be represented by (0,1)-sequences with as many 0's as 1's, for which each truncated sequence has more 0's than 1's. These sequences are not difficult to count, whereas counting the trees directly looks very complicated. The problem of counting the number of dissections of an n-gon into triangles looks quite different. Usually one first discovers that this problem leads to the same answer as the previous one before realizing that it can be represented by the same kind of (0,1)-sequences.

EXAMPLE 4. The reverse situation is also useful as an example. For instance, a problem on (0,1)-matrices can look like a difficult abstract problem. Interpreting the matrix as a representation of some combinatorial object translates the question into other terminology and can make it much easier.

Counting techniques.

This topic includes double counting, the principle of inclusion and exclusion, Möbius inversion, the use of quadratic forms, one-to-one mappings, generating functions, Polya theory and probablistic methods. Again a few (favorite) examples.

EXAMPLE 5. This is one of the problems that students try to solve with no tools. Let the edges of a complete graph on six vertices be colored red and blue in some way. Prove that there is a triangle with all three edges of the same color (a monochromatic triangle). Nearly all the students give the same proof. From any vertex there must be three edges with the same color, say red. The three edges between the other endpoints of the red edges are either all blue or one of them is red and in both cases we have a monochromatic triangle. So far, so good. The second question is to show that there are actually at least two monochromatic triangles. This yields three possible solutions : the empty one, complete nonsense, or a several page case analysis that is actually correct. Then comes double counting in class! Every non-monochromatic triangle has two vertices where a red and a blue edge meet; call this a red-blue V. Clearly every vertex yields at most six of these red-blue V's. So, this second way of counting (or better estimating) the number of non-momochromatic triangles shows that there are at most 18 of them. As K_6 contains 20 triangles, we are done in a few lines.

EXAMPLE 6. After the usual examples of inclusion-exclusion it is useful to point out the reverse procedure. Try to prove the formula

$$\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n = \begin{cases} n! & \text{if } k=n, \\ 0 & \text{if } k>n. \end{cases}$$

This can be done using analysis but it is not trivial. The term $(-1)^i$ in the sum suggests that maybe something was counted using inclusion and exclusion. What? This takes some thinking. The answer is : the number of surjections from an n-set to a k-set and the formula becomes a triviality.

EXAMPLE 7. The following quadratic form method occurs in very many different situations. Let a_i denote the number of combinatorial objects of a certain kind that have exactly i whatevers. Often one can easily count pairs of whatevers. Since $\sum a_i$ counts the number of objects in question, $\sum ia_i$ counts the number of whatevers, and finally $\sum\binom{i}{2}a_i$ counts pairs of whatevers, one can calculate expressions of the form $\sum(i-m)(i-m-1)a_i$, where the choice of m is free. The fact that this quadratic form is nonnegative yields an inequality. It is surprising how often this idea is used in combinatorics without pointing out that it is a general method.

(Non-)existence and constructions.

Methods to be treated here include counting (probabilistic methods), the method of descent or minimal counterexample, algorithms and search techniques, induction and recursion, product techniques, substitution, algebraic methods, contraction, introducing extra structure. We give a few examples.

EXAMPLE 8. The construction of a Latin square of order mn from one of order m and one of order n is very similar to the construction of a Hadamard matrix of order mn from one of order m and one of order n. Both constructions should occur. Later one can use similar product methods in the construction of block designs. Even the idea of the product of graphs is analogous.

EXAMPLE 9. A well known proof technique in number theory can be extended to several parts of discrete mathematics, such as graph theory. To prove a theorem on finite configurations one assumes that it is not true, i.e. a counterexample exists. In that case there exists a *minimal* counterexample, where minimal refers to the number of components that justify the word "finite". One has to think of a way of reducing this number (delete a vertex or replace the integer n by $n-1$) in such a way that the reduced object is still a counterexample. This yields a contradiction and thus the theorem is proved. Again, the point of this talk is that if one decides to show an example of the method, one should show *several* rather different examples.

EXAMPLE 10. The idea of *substitution* occurs in many constructions. Examples are replacing a vertex of a graph by some graph, points of a configuration by n-gons (e.g. in Joyal theory), and the following. In a block design with blocks of different sizes (every pair of points is in λ blocks) let there be a block B with seven points. We delete B and replace it by the seven triples (lines) of the Fano-plane (a (7,3,1)-design). The $\binom{7}{2} = 21$ pairs that were covered by B are now covered by the seven lines of the plane. This method is used to replace the difficult restriction of constant blocksize by freedom in that respect in the first round of a construction, followed by substitutions of the the type mentioned above to achieve a prescribed constant blocksize.

EXAMPLE 11. Assume that a combinatorial object is defined by combinatorial restrictions only. It may be difficult to construct even one example of such an object. One can freely introduce extra structure, such as symmetry, an automorphism group, etc. in order to force the construction in a certain direction. If the extra requirements are not already prohibitive, one may have an easy construction of a first example of the theory. Again, this is a *princple* that should be illustrated by examples!

Applications.

Discrete mathematics as a course should be full of examples of applications in a wide area of subjects. Students should not only learn a number of applications but should recognize situations where a certain part of discrete mathematics is the *natural tool* to use. One should move from computer science to social sciences to electrical engineering to design of experiments, etc. Examples may be elementary, obvious, everyday, but it is essential to have several others that ensure that the students *enjoy* the course. These are surprising, challenging, ingenious (like Instant Insanity), recent (such as satellite communication or the compact disc). Again, a few favorites as examples.

EXAMPLE 12. Suppose one has a standard nonerasable binary memory such as paper tape (or a compact disc). Assume that one wishes to store one of the integers 1 to 7 in this memory on four consecutive occasions. The usual procedure is to reserve twelve bits for this purpose, where the four consecutive triples each take care of one storage of a binary 3-tuple. The world supply shortage has now reached the stage where we cannot afford this and have to achieve the same with a memory of only seven bits! (The reader should try to prove as an exercise that it is not possible to solve the storage problem with a memory of six bits.) The solution is provided by the Fano plane, a finite geometry with seven points and seven lines, three points to a line and three lines through a point, any two points on a unique line. Number the points 1 to 7 and on the first storage let a 1 in position i indicate a storage of the integer i. This is still easy. The next step is not difficult either. If the memory contains a 1 in position i and one wishes to store the integer j as new information, find the *unique* line through i and j and if k is its third point, put a 1 in position k. The reading device for this binary memory is told that if it sees two 1's, then it should interpret these as "the third point of the corresponding line". Two more usages of this memory to go and we leave it as an exercise to decide how to do it (hint : a change of memory with two 1's results in four 1's; a subsequent change leads to either five or six 1's).

EXAMPLE 13. During the treatment of Hadamard matrices one has given the product construction and therefore the trivial Hadamard matrix of order two (rows ++,respectively +−) makes it possible to construct such matrices of order 2^n. As an exercise the students have shown that this leads to a matrix H of order 32 with the property that there are six columns in the array consisting of H and $-H$ such that the corresponding 64 rows in this array are all different in these six columns (note that $2^6 = 64$). As application one treats the transmission to earth of pictures of Mars by the stellite Mariner '69. A picture is divided into very little squares (pixels) and for each square the degree of blackness is measured in a scale of 0 to 63 (expressed in binary). In this way the picture results in a long sequence of 0's and 1's to be transmitted to earth. The transmitted sequence is corrupted by noise and the effect is that the receiver sometimes interprets a 0 as a 1 and vice versa. In practice there was so much noise that pictures would have been completely useless. Suppose we are willing to take roughly five times as long to transmit a picture. We could repeat each bit five times; if no more than two out of five are received incorrectly, the receiver makes the right choice. This would be a substantial improvement but what was done in practice in 1969 was very much better. An integer, say 43, in binary 101011 was changed to the corresponding sequence of +'s and −'s (i. e. $+-+-++$) and transmitted as the corresponding row of 32 +'s and −'s of the array of H and $-H$. This also takes five times as long (roughly). The reader should convince himself that as many as seven of the transmitted symbols may be received incorrectly and nevertheless the receiver will still have the correct row as the most likely one. The result is known : the pictures were of great quality. A true and recent example!

Reference.

The ideas presented in this talk were used as guiding principle in the book *A course in combinatorics* by J. H. van Lint and R. M. Wilson, Cambridge University Press, 1992.

The talk by A. Revuz in Utrecht appeared as "Les pièges de l'enseignement mathématiques",*Educational Studies in Mathematics* 1 (1968), 31–36.

J. H. van Lint, Eindhoven University of Technology, The Netherlands, August 1992.