COMBINATORICS

Takeshi Tokuyama
Tohoku University, Sendai, Japan

Keywords: Combinatorics, Set Theory, Discrete Geometry, Partition Theory, Matroid, Hypergraph, Codes

Contents

Summary

The art of counting and computation is the origin of mathematics. For example, if we want to count a heap of stones, we usually group them into clusters each of which contains ten stones: the method is strongly related to the idea of decimal system.

The mathematics on sophisticated art of counting and discrete computation is called *combinatorics*. In short, combinatorics is an art to represent discrete objects

systematically and comprehensively.

Combinatorics is in the heart of computer science, since we cannot handle large data without the art of efficient discrete computation. Indeed, every data set in modern computers is digital (thus, discrete) and represented by using combinational structures.

Therefore, the current IT (Information Technology) society could not exist without mathematical works on combinatorics. Moreover, combinatorics is applied to several sciences/engineering such as physics, chemistry, life science, graphics, architecture, and even social sciences and arts.

We start with the problem of counting poker hands, which needs elementary combinatorics on sequences and set systems. Then we survey fundamental combinatorial structures such as graphs, hypergraphs, codes, designs, and matroids. In addition, we introduce advanced classical topics including Ramsey theory, combinatorial geometry and partition theory.

1. Introduction

Combinatorics is the mathematical study of finite sets and discrete structures, such as set systems, sequences, graphs, hypergraphs, matroids, points and lines in the plane, planes in the space, and polytopes.

The following are typical problems in combinatorics:
Example 1 (Enumeration of poker hands). Consider the number of different hands in a poker game; this is the number of *combinations* of 5 cards from 52 cards, which is 2598960. Among them, the number of hands with a *pair* (two same ranked cards) is 1281072. Moreover, the number of hands with "no value" in the poker game is 1302540. How to compute these numbers?

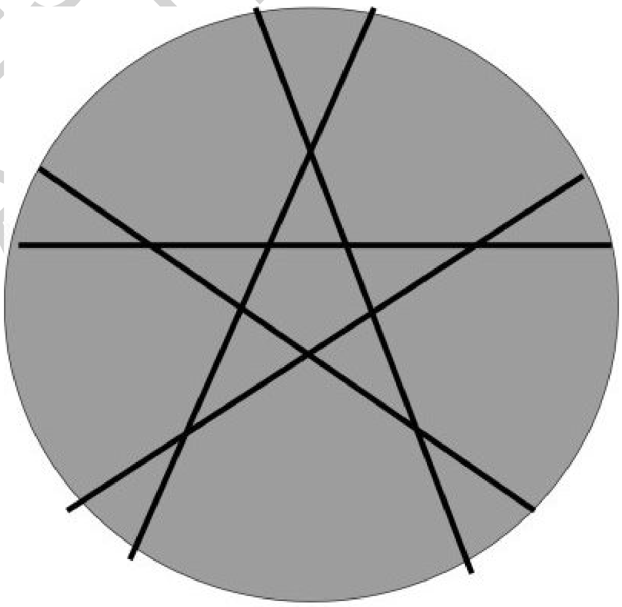

Figure 1: Cutting a disk with five lines.

Example 2 (Cake cutting problem). If we cut a disk (a cake) with 5 lines (knife-cuts), the disk can be cut into 16 pieces (Figure1). Is 16 the maximum number of pieces? What is the number if we cut with 10 lines?

Example 3. If we place 9 points in the plane such that no three points are on a line, we can always find 5 points forming a convex 5-gon (pentagon) (Figure 2). Why does this property hold? However, the configuration of Figure 2 does not contain a point set forming a convex (hexagon)6-gon. How many points are necessary so that they always contain a convex 6-gon?

Example 4 (Kirkman's fifteen schoolgirls problem). Fifteen young ladies in a school walk out three abreast for seven days in succession: it is required to arrange them daily, such that no two shall walk twice abreast. Is it really possible to do such an arrangement?

The area of combinatorics is very wide [3], and it is applied to every field of modern mathematics. In particular, probability theory [1] and group theory [8, 9, 10] are strongly related to combinatorics. Moreover, combinatorics is a fundamental tool in computer science [4, 5], operations research [7], theoretical physics, and mechanical engineering.

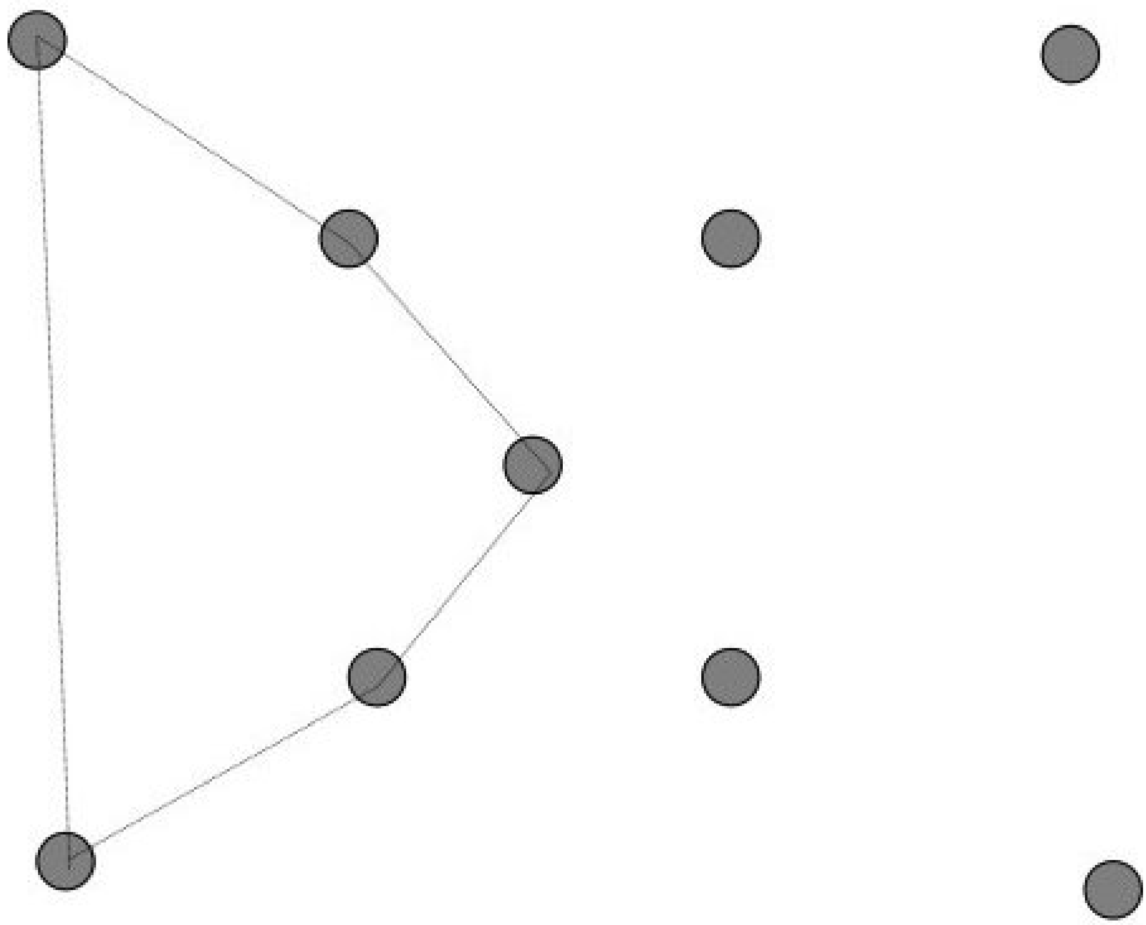

Figure 2: Any 9 points must contain a convex 5-gon

1.1. Set Systems and Bit Sequences

Given a set U of n elements, $\mathcal{P}(U)$ is the set of all subsets of U. For example, if $U = \{1, 2\}$, $\mathcal{P}(U)$ consists of four sets $\emptyset, \{1\}, \{2\}$, and $\{1, 2\}$, where $\emptyset$ means the empty set. $\mathcal{P}(U)$ is often called the *power set* of U. A subset of $\mathcal{P}(U)$ is called a *set system*, and U is called its underlying set. $\{\{1\}, \{1, 2\}\}$ is an example of a set system.

A subset S of $U = \{1, 2, ..., n\}$ corresponds to a *bit sequence* (i.e. sequences of 0 or 1) of length n: we assign 1 to the *i*-th entry of the sequence if $i \in S$, otherwise 0. For example, bit sequences 00, 10, 01, and 11 corresponds to $\emptyset, \{1\}, \{2\}$, and $\{1, 2\}$ in $\mathcal{P}(\{1, 2\})$, respectively. Thus, we have the first important observation that the number $|\mathcal{P}(U)|$ of different subsets in U is 2^n, since there are 2^n bit sequences of length n.
In a computer, everything is represented by using bit sequences. For example, alphabets are usually represented by using bit sequences of length 8. There are $2^8 = 256$ different bit sequences of length 8, which corresponds to elements of $\mathcal{P}(U)$ for $U = \{1, 2, ..., 8\}$. The set of 26 alphabets can be realized as a subset of $\mathcal{P}(U)$, and hence forms a set system called a *bite code* of alphabets. Thus, the study of set systems is quite fundamental in computer science.

1.2. Level Set and Combination Numbers

We refine $\mathcal{P}(U)$ and consider the set system $\mathcal{P}_k(U)$ of all subsets of U containing exactly k elements. This is called the *level set* of rank k in $\mathcal{P}(U)$. Since each hand of the poker game consists of five different cards, the set of hands in the poker game is $\mathcal{P}_5(U)$ for the set U of 52 different cards (we exclude Jokers).

The cardinality of $\mathcal{P}_k(U)$ is called the *combination number* of choosing k elements from n elements, and denoted by ${}_nC_k$, which is the most basic and popular tool in combinatorics. Therefore, the number of different hands in the poker game is ${}_{52}C_5$.

Let us compute ${}_4C_2$ for getting intuition. We can see that there are 6 subsets of size 2 in $\{1, 2, 3, 4\}$: $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \text{and} \{3, 4\}$. Thus, ${}_4C_2 = 6$. Let us consider it as follows: instead of considering subsets, we consider the sequences of length 2 consisting of two distinct elements of $\{1, 2, 3, 4\}$. The first entry can be arbitrary chosen (hence we have 4 choices), and the second entry can be chose from 3 other elements than the first choice. Hence, there are $4 \times 3 = 12$ such sequences. Since we can create two sequences from each of combinations of two numbers, for example, $1, 2$ and $2,1$ from {1, 2}, we can show that $12/2 = 6$ is the number of subsets of size 2 in $\{1, 2, 3, 4\}$.

This argument holds in general: A sequence of length k consisting of distinct elements of a set S is called a *k-element sequence* of S. A *k*-element sequence is called a *permutation* if it uses all elements of S (i.e. $k = |S|$). Analogous to the argument above, we can observe that the number of different *k*-element sequences of U is $n\times(n-1)\times(n-2)\times...\times(n-k+1)=\prod_{i=n-k+1}^{n} i.$ This number can be represented as $n!/(n-k)!$ if we use the *factorial function* $n!=\prod_{i=1}^{n} i,$ where $0!$ is defined to be 1. On the other hand, $k!$ different permutations can be created from a combination of k numbers. Hence, we have the following two *factorial expressions* of the combination number:

$$ {}_nC_k = (\prod_{i=n-k+1}^{n} i)/k! $$

$$ {}_nC_k = n!/(n-k)!k!. $$

By using the first expression, we can compute ${}_{52}C_5 = 52\times51\times50\times49\times48/5! = 2598960.$

Since $|\mathcal{P}(U)| = \sum_{k=0}^{n} |\mathcal{P}_k(U)|,$ we have $\sum_{k=0}^{n} C_k = 2^n$. The following useful formulas can be directly derived from the factorial expressions:

$ {}_nC_k = {}_nC_{n-k}$ Symmetry of combination numbers
$ {}_nC_k = (n/k)\,{}_{n-1}C_{k-1}$ Recursive formula
$ {}_nC_k = {}_{n-1}C_k + {}_{n-1}C_{k-1}$ Pascal's formula (two-dimensional recursive formula)

1.3. Inclusion-Exclusion Principle

Given a property a on a set S consisting of N elements, let $N(a)$ be the number of elements which satisfy a. The negation of a property a is denoted by $\neg a$; Hence, $N(\neg a)$ is the number of elements which do not satisfy a. If we have two properties a and b, $N(a\wedge b)$ is the number of elements which satisfy both a and b, and $N(a\vee b)$ is the number of elements which satisfy either a or b. Then, it is not difficult to check the following three formulas:

$$ N(\neg a) = N - N(a) $$

$$ N(a\vee b) = N(a) + N(b) - N(a\wedge b) $$
$$ N(\neg a\wedge \neg b) = N - N(a) - N(b) + N(a\wedge b) $$

The above three formulas are basic cases of the principle of inclusion-exclusion. More generally, given a set I of properties, $N(\wedge I)$ is the number of elements that satisfy all of the properties. The expression $\neg I$ denotes the set of negated properties of I. The principle of inclusion-exclusion is the following:

$$N\left(\wedge(\neg I)\right) = \sum_{0 \neq J \subset I} (-1)^{|J|} N\left(\wedge J\right).$$

In the poker game, we consider the set $S = \mathcal{P}_5(U)$ of all possible hands, where U is the set of 52 cards consisting of four suits each consists of 13 cards ranked by numbers A, 2,…, 10, J, Q, and K. Here, A, J, Q, and K represent 1, 11, 12, and 13, respectively. We consider the property *pair* that “the hand has at least a pair of same ranked cards”. We compute $N(pair)$ by using the inclusion-exclusion property $N(pair) = N - N(\neg pair)$, where N is the number of all possible hands and we have already seen that N = 2598960. A hand satisfying $\neg pair$ must consist of cards with 5 different numbers. Therefore, to compute $N(\neg pair)$, we consider the combinations of 5 numbers from 13, and assign arbitrary a suit to each card.

Hence, $N(\neg pair) = {}_{13}C_5 \times 4^5 = 1287 \times 1024 = 1317888$, and $N(pair) = N - 1317888 = 2598960 - 1317888 = 1281072$.

A poker hand is called *straight* if it forms a consecutive sequence of numbers (10JQKA is permitted but JQKA2 is not) and *flush* if it consists of cards with a single suit.

$N(\neg pair \wedge \neg straight \wedge \neg flush)$ is the number of hands with “no value” in the poker game. By using the inclusion-exclusion principle, $N(\neg pair \wedge \neg straight \wedge \neg flush) =$
$N - N(pair) - N(straight) -$
$N(flush) +$
$N(straight \wedge flush) + N(pair \wedge straight) + N(pair \wedge flush) - N(pair \wedge straight \wedge flush)$

It can be observed that a hand with a pair can be neither straight nor flush. Hence, $N(pair \wedge straight) = N(pair \wedge flush) = N(pair \wedge straight \wedge flush) = 0.$ If we ignore suits, a straight hand is a sequence starting from A, 2, 3,…, or 10. Hence, there are 10 such sequences and we can assign an arbitrary suit to each five cards. Thus, $N(straight) = 10 \times 4^5 = 10240.$ We can easily see that $N(flush) = 4 \times {}_{13}C_5 = 5148,$ and $N(straight \wedge flush) = 40.$ Hence, $N(pair \wedge straight \wedge flush) =$ $2598960 - 1281072 - 10240 - 5148 + 40 = 1302540.$

1.4. Recursive Formulas and Asymptotic Bounds

In combinatorics, we often want to compute a positive valued function $f(n)$ dependent on a natural number n. A typical technique to compute $f(n)$ is to find a *recursive formula* and solve it. A recursive formula is a formula expressing $f(n)$ in terms of $f(k)$ for $k \leq n-1$.

Consider the problem of cutting a disk (a cake) with n lines (knife cuts) and counting the maximum number $f(n)$ of connected pieces (the *cake cutting* problem). We can easily check that $f(1) = 2$ and $f(2) = 4.$ It seems that $f(n) = 2^n$ is a good candidate for the answer, although it is wrong since $f(3)=7$ (Figure 3).

Suppose that we have cut the disk with $n-1$ lines into (at most) $f(n-1)$ pieces. If we give a new cut with a line l, the line l has $n-1$ intersecting points with the existing $n-1$ lines, and hence l is cut into n segments with the existing lines. If the cake is sufficiently large, the line l cuts n pieces of the current subdivision, since a segment on l must be created if we cut a piece in the current subdivision with l. Therefore, we have a recursive formula $f(n)=f(n-1)+n.$ From this recursive formula, we have $f(n)=n+(n-1)+(n-2)+\ldots+1+f(0)=(n(n+1)/2)+1.$ An equivalent but more beautiful expression is $f(n)={}_nC_2+{}_nC_1+{}_nC_0$. For example, $f(5)$ equals 16, and $f(10)$ equals 56.

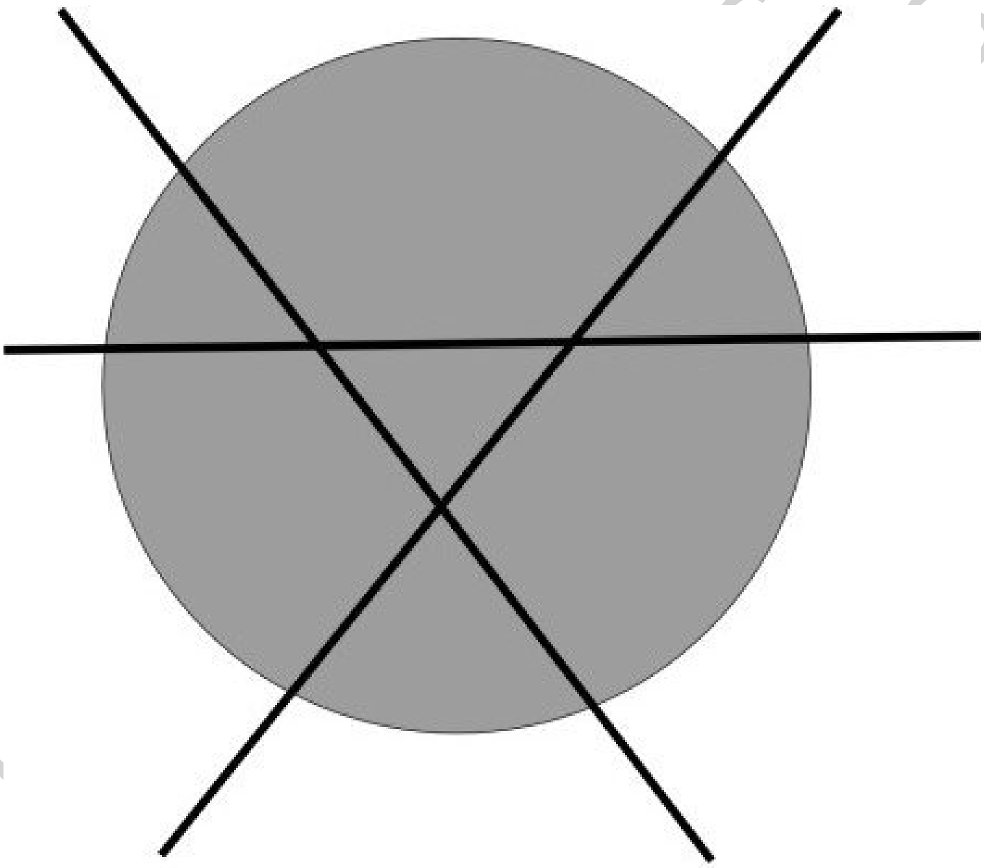

Figure 3: Cutting a disk with three lines.

Functions with more than one variable can have multivariate recursive formulas: for example, the combination number $f(n,k)={}_nC_k$ satisfies a recursive formula $f(n,k)=f(n-1,k-1)+f(n-1,k)$ (Pascal’s formula).

Recursive formulas are sometimes difficult to solve, or have ugly solutions. In such a case, we consider the *asymptotic behavior* of a function; that is, how the function grows when n becomes very large. To describe the asymptotic behavior of a function $f(n)$, we use another familiar function $h(n)$ and estimate $f(n)$ by using the Big-O and Big-Ω notations $O(h(n))$ and $\Omega(h(n))$.

The expression $f(n)=O(h(n))$ means that there exist constants c and N such that $f(n)<c\cdot h(n)$ holds for every n satisfying $n>N$. The expression $f(n)=\Omega(h(n))$ means that there exist a constant c' such that for any constant N there exists an $n>N$ such that $f(n)<c'\cdot h(n)$. We say $h(n)$ gives an asymptotic upper bound (resp. a lower bound) of $f(n)$ if $f(n)=O(h(n))$ (resp. $f(n)=\Omega(h(n))$.

To obtain an asymptotic upper bound of a function $f(n)$, we can use recursive formulas given as inequalities. The following examples are basic asymptotic bounds obtained from recursive inequalities:

1. $f(n)\le 2f(n/2)+n^c$ for a constant c gives: $f(n)=O(n^c)$ if $c>1$, $f(n)=O(n\log n)$ if $c=1$, and $f(n)=O(n)$ if $c<1$.
2. $f(n)\le f(n-1)+n^c$ gives: $f(n)=O(n^{c+1})$ if $c\ne -1$, $f(n)=O(\log n)$ if $c=-1$.
3. $f(n)\le f(cn)+n$ for a constant $c<1$ gives $f(n)=O(n)$.
4. $f(n)\le cf(n-1)$ gives $f(n)=O(c^n)$.

In each of the above formulas, if we reverse the inequality, we obtain a lower bound instead of the upper bound.

-
-
-

Bibliography

[1] Alon N., Spencer J. H., and Erdős P. (1991) *The Probabilistic Method*, 254 pp, John Wiley & Sons. [Textbook on probabilistic methods in combinatorics.]

[2] Andrews, G.E. (1976). *The Theory of Partitions*, Encyclopedia of Mathematics and Its Applications 2, 255 pp, Addison Wesley. [Systematic survey of partition theory and related number theory.]

[3] Graham R.L, Grötschel M., and Lovász, L.(ed.) (1995). *Handbook of Combinatorics*, two volumes, 2198 pp, MIT Press. [Complete handbook of combinatorics, including theory and applications.]

[4] Graham R.L., Knuth D.E., and Patashnik O. (1994) *Concrete Mathematics* (2nd ed.), 657 pp, Addison-Wesley. [Elementary book to give basic combinatorial results.]

[5] Knuth, D. E. (1997, third edition) *The Arts of Computer Programming* 3 Volumes, Addison-Wesley.

[A classical book on theory of computing, including many combinatorial ideas for efficient computation in computer science/engineering.]

[6] Pach J. and Agarwal P.K. (1995). *Combinatorial Geometry*, 354 pp, Wiley & Sons. [This presents a comprehensive survey of combinatorial geometry.]

[7] Roberts F. S. (1984). *Applied Combinatorics* 606 pp, Prentice Hall. [Applications of combinatorics to operations research are comprehensively explained.]

[8] Stanley, R.P. (1997) *Enumerative Combinatorics I*, Cambridge University Press. [The first volume of a complete book series on enumerative combinatorics. Basic theory on posets and applications to algebra.]

[9] Stanley R.P. (1999) *Enumerative Combinatorics II* 581 pp, Cambridge University Press. [The second volume of a complete book series on enumerative combinatorics. Advanced theory on generating functions.]

[10] Weyl, H. (1946) *The Classical Groups* 320 pp, Princeton University Press. [Classical book to characterize group representations by using Young diagrams.]

Biographical Sketch

Takeshi Tokuyama was born January 22, 1957. He received BS, MS and PhD degrees in 1979, 1981, and 1985 all from the University of Tokyo. From December 1986 to August 1999 he was Research Staff Member, IBM Tokyo Research Laboratory. From April-August 1999 he was Guest Professor at the University of Tokyo. Since September 1999 he is with the Tohoku University as a Professor.