УДК 519.85(023)+372.8:51

КЛАССИЧЕСКИЕ КОМБИНАТОРНЫЕ ОБЪЕКТЫ НА СОРЕВНОВАНИЯХ ПО ПРОГРАММИРОВАНИЮ

М.И. Киндер[1]

1 *detkinm@gmail.com*; Казанский федеральный университет

В статье обсуждается рекурсивный подход к перечислению некоторых классов комбинаторных задач. Классические комбинаторные объекты — частые гости олимпиадных соревнований различного уровня. Комбинаторные проблемы, в которых они возникают, опираются на зависимость от рекуррентных соотношений и поэтому, чаще всего, решаются с помощью метода динамического программирования. При таком подходе сложные задачи решаются путём разбиения их на более простые и мелкие проблемы. Большинство примеров в этой статье встречались на соревновании по спортивному программированию — Открытом кубке им. Е.В. Панкратьева (Гран-При Татарстан). Это соревнование ежегодно проводится в г. Казани и служит одним из этапов подготовки студенческих и школьных команд для участия в финале ACM ICPC и Всероссийской олимпиады школьников по информатике. Полные тексты всех этих задач доступны в Интернете: www.icl.ru/turnir.

Ключевые слова: олимпиады по спортивному программированию, комбинаторные объекты, динамическое программирование.

1. Введение

Соревнования по информатике были введены около сорока лет назад с прекрасной идеей привлечения талантливых молодых людей к компьютерным наукам и программированию. За прошедшие годы эти олимпиады продемонстрировали свою эффективность в отборе талантов и формировании высококвалифицированных специалистов в сфере компьютерных технологий.

Начиная с 2000 года, российская IT-компания ОАО «ICL-КПО ВС» проводит ежегодный «Турнир ICL» — открытый чемпионат Татарстана по спортивному программированию среди команд школьников и студентов. С 2010 года лучшие команды и их руководители получают премии Президента Республики Татарстан. В 2012 году этот турнир стал одним из этапов Открытого Кубка имени Е.В.Панкратьева и приобрёл статус международного соревнования. Правила проведения Турнира ICL близки к правилам командного чемпионата мира по программированию ACM ICPC среди студентов. Команда, состоящая из трех человек, имеет только один компьютер и пять часов времени, чтобы решить от 10 до 12 довольно сложных задач на одном из разрешенных на турнире языков программирования. Успешное решение задач помимо знания языков программирования предполагает математическую подготовку, знание алгоритмов и структур данных, а также слаженную командную работу в условиях дефицита времени.

Каждый год студенческие команды-финалисты чемпината мира по спортивному программированию и их ближайшие конкуренты традиционно собираются в летних и зимних тренировочных лагерях, организованных специально для них в г. Петрозаводске и г. Ижевске, куда, кроме российских программистов, приезжают также команды из Польши, Чехии, Японии, Украины и Беларуси. Кроме того, несколько открытых чемпионатов проводится в течение года, где также можно встретиться с потенциальными финалистами ACM ICPC. Среди них упомянем Открытую Всесибирскую олимпиаду по программированию им. И.В. Поттосина, Открытый чемпионат Республики Татарстан (Турнир ICL) по спортивному программированию и чемпионат Урала.

В этой статье мы разберём несколько комбинаторных задач Турнира ICL, одного из этапов Открытого кубка им. Е.В.Панкратьева (Гран-При Татарстан). Полные тексты всех этих задач доступны в Интернете по адресу www.icl.ru/turnir.

2. Избранные задачи Турнира ICL

Идеи олимпиадных задач могут возникнуть из различных математических дисциплин, однако некоторые разделы математики особенно полезны для создания задач по олимпиадной информатике. В этом параграфе мы разберём несколько задач комбинаторного характера, которые встречались на командном турнире ICL по спортивному программированию. Комбинаторные проблемы опираются на зависимость от рекуррентных соотношений и поэтому, чаще всего, решаются с помощью метода *динамического программирования*. При таком подходе сложные задачи решаются путём разбиения их на более простые подзадачи [1].

Рассмотрим, например, следующую задачу.

Беспорядки

[Муниципальный этап Всероссийской олимпиады школьников по информатике, Республика Татарстан, 2013.]

На книжной полке известного учёного Василия Шекспирова беспорядочно стоят n *книг из его полного собрания сочинений. Каждый том имеет свой уникальный номер — целое число от* 1 *до* n. *В целях научности и систематичности Вася решил оценить степень беспорядка расставленных книг. С его точки зрения два тома образуют беспорядок, если том с меньшим номером стоит правее тома с бóльшим номером. Степенью беспорядка учёный назвал общее число* k *всех беспорядков в расстановке. Например, в расстановке* $(2,1,5,3,4)$ *беспорядки образуют пары томов* $(2,1)$, $(5,3)$ *и* $(5,4)$, *поэтому степень беспорядка такой расстановки равна* 3. *Василий заметил, что есть и другие расстановки из* 5 *книг с той же степенью беспорядка* 3, *— например,* $(2,1,4,5,3)$, $(1,3,4,5,2)$ *и другие.*

Ваша задача — помочь Василию Шекспирову подсчитать общее число расстановок из n *книг с заданной степенью беспорядка* k $(2 \le n \le 10000; 0 \le k \le 30)$. *Ответ записать по модулю* (10^9+9).

Приведем решение, основанное на идее динамического программирования. Пусть $d[n,k]$ — количество расстановок из n книг с заданной степенью беспорядка k. Докажем, что для функции $d[n,k]$ справедливы следующие рекуррентные соотношения:

$$d[n,k] = d[n,k-1] + d[n-1,k] - d[n-1,k-n]$$

с начальными условиями $d[n,0] = 0$, $d[n,1] = n-1$ и $d[n,k] = 0$ при $k < 0$. Действительно, пусть перестановка $(a_1, a_2, \ldots, a_n)$ содержит k инверсий и $n > k$. Множество всех перестановок порядка n разобьем на два класса — те, у которых $a_n = n$, и те, у которых $a_i = n$, где $i < n$. Элементы первого класса являются, по сути, перестановками порядка $n-1$, поэтому количество таких перестановок равно $d[n-1,k]$. Для перестановок второго класса мы можем поменять местами $a_i = n$ и a_{i+1}, получив перестановку порядка n с $k-1$ инверсиями.

Обратно, в любой перестановке порядка n с $k-1$ инверсиями первый элемент отличен n, так как иначе количество инверсий будет не менее $n-1 \geq k$. Значит, поменяв местами элемент n с предыдущим, мы получим перестановку с k инверсиями. Следовательно, количество перестановок во втором классе равно $d[n,k-1]$. Отсюда при $n > k$ получаем $d[n,k] = d[n,k-1] + d[n-1,k]$. Доказательство рекуррентного соотношения при $n \leq k$ можно провести аналогично (разбивая множество всех перестановок на n классов в зависимости от положения элемента n).

Теперь осталось организовать рекурсивную функцию `d(n,k)`, сохраняя все ранее подсчитанные значения функции в двумерном массиве `table`[1..10000, 0..30]:

```
function d(n: longint; k: longint): int64;
begin
  if  k < 0  then  begin d := 0;     exit; end;
  if  k = 0  then  begin d := 1;     exit; end;
  if  k = 1  then  begin d := n - 1; exit; end;
  if  (k > n * (n - 1) div 2)  then begin d := 0; exit; end;
  if  table[n,k] > -1 then begin d := table[n,k]; exit; end;
  Result := (d(n - 1,k) + d(n,k - 1) - d(n - 1,k - n)) mod 1000000009;
  table[n,k] := Result;
end;
```

Вызов этой функции из основной программы:

```
begin fillchar(table, sizeof(table), -1);
write(d(n,k));
end.
```

Прочные штабеля

[Открытый Кубок имени Е.В. Панкратьева, 2014, автор — *М. Киндер.*]

В соответствии с технологической документацией каждый штабель формируется в точности из k коробок путем установки их друг на друга. Каждая коробка весит целое число килограммов, вес штабеля должен составлять n килограммов. В целях обеспечения прочности и во избежание эффекта «расплющивания» вес каждой коробки должен быть не меньше суммарного веса всех находящихся над ней коробок. Например, для $n = 7$ существуют два прочных штабеля из $k = 3$ коробок: $7 = 1+2+4$ и $7 = 1+1+5$. (В каждом равенстве первое слагаемое — это вес верхней коробки, второе — вес второй сверху коробки, и, наконец, третье слагаемое указывает вес нижней коробки.)

Всем штатным программистам необходимо подсчитать общее количество способов формирования указанных штабелей и довести до сведения главного технолога. Ответ записать по модулю (10^9+9).

Для решения задачи сначала уточним понятие «расплющенности». Разбиение натурального числа $n = p_1+p_2+\ldots+p_k$ назовём *прочным* (*non-squashing partitions* [2]), если целые слагаемые p_i удовлетворяют условиям $1 \le p_1 \le p_2 \le \ldots \le p_k$, причём каждая следующая часть не меньше суммы предыдущих, то есть $p_1 + p_2 + \ldots + p_i \le p_{i+1}$ для всех i от 1 до $k-1$. Штабель из k коробок, у которых веса (начиная с верхней) равны p_1, p_2, …, p_k, будет прочным только в том случае, когда разбиение числа n на k слагаемых p_i является прочным.

Пусть $m[n,k]$ — количество прочных разбиений натурального числа n на k слагаемых. Несложно доказать, что числа $m[n,k]$ удовлетворяют следующим рекуррентным соотношениям:

$$m[2n,k] = m[2n-1,k] + m[n,k-1],$$
$$m[2n+1,k] = m[2n,k]$$

с начальными условиями $m[n,0] = 0$ и $m[n,1] = 1$, а также $m[n,k] = 0$ для всех $k > n$. Действительно, докажем сначала второе соотношение. Пусть $2n+1 = p_1 + p_2 + \ldots + p_k$ — прочное разбиение числа $2n+1$ на k частей. Очевидно, что бо́льшая часть p_k этого разбиения больше n, в противном случае

$$2n+1 = p_1 + p_2 + \ldots + p_{k-1} + p_k \le p_k + p_k \le 2n,$$

получаем противоречие. Но тогда сумма оставшихся частей $p_1 + p_2 + \ldots + p_{k-1}$ меньше n, то есть числа p_k и $p_1 + p_2 + \ldots + p_{k-1}$ отличаются не менее, чем на 2. Поэтому мы можем вычесть 1 из части p_k , получив при этом прочное разбиение числа $2n$ на k частей. Таким образом, мы установили взаимно однозначное соответствие между этими двумя множествами, поэтому $m[2n+1,k] = m[2n,k]$.

Теперь докажем второе соотношение. Как и в первом случае, доказывается, что бо́льшая часть p_k разбиения не меньше n, а сумма остальных слагаемых разбиения не больше n. В случае $p_k > n$ мы можем вычесть 1 из части p_k, получив при этом прочное разбиение $(p_1, p_2, \ldots, p_k - 1)$ числа $2n-1$. И наоборот, добавив 1 к наибольшей части прочного разбиения числа $2n-1$, получим прочное разбиение числа $2n$ с наибольшим слагаемым $p_k > n$. Значит, количество прочных разбиений, в которых бо́льшая часть p_k больше n, равно $m[2n-1,k]$. Если же $p_k = n$, то оставшиеся части $p_1, p_2, \ldots, p_{k-1}$ в сумме дают n, и значит, образуют прочное разбиение числа n. Следовательно, $m[2n,k] = m[2n-1,k] + m[n,k-1]$.

Доказанные рекуррентные соотношения позволяют вычислить значение $m[n,k]$. Детализация этого алгоритма уже не представляет особого труда.

Онлайн-энциклопедия целочисленных последовательностей [4], по сути, является неисчерпаемым источником для интерпретации классических и других комбинаторных объектов, удовлетворяющих различным рекуррентным соотношениям; даже простой просмотр этой энциклопедии может помочь созданию интересных олимпиадных заданий. Следующая задача — еще одна интепретациия известных комбинаторных чисел Моцкина.

Разреженные скобочки

[Открытый Кубок имени Е.В.Панкратьева, 2015, автор — *М. Киндер*.]

Порядок вычислений в арифметических выражениях задается расстановкой скобок, например, $(3 \cdot (2+1)) \cdot (4-5)$. *Если удалить все элементы выражения, за исключением* *скобок, то оставшиеся символы образуют скобочную последовательность* $(())()$. *Предположим теперь, что кроме открывающихся и закрывающихся скобок может присутствовать ещё один символ, не нарушающий правильность скобочных последовательностей. Например, символ «*0*». Такую последовательность будем называть разреженной скобочной последовательностью. Понятие разреженной скобочной последовательности можно определить и так:*

- *Пустая строка считается разреженной скобочной последовательностью.*
- *Если S и T — разреженные скобочные последовательности, то строки $0S$, $S0$, (S) и ST также являются разреженными скобочными последовательностями.*

Глубиной разреженной скобочной последовательности называется максимальная разность между количеством открывающих и закрывающих скобок в префиксе последовательности. (Префиксом строки S называется строка, которую можно получить из S удалением некоторого количества символов с конца строки. Например, префиксами строки «$ABCAB$» являются строки «», «A», «AB», «ABC» и «$ABCA$» и «$ABCAB$».) Так, глубина последовательности «$(0)(0())0$*» равна двум (префикс «*$(0)(0($*» содержит три открывающие и одну закрывающую скобки).*

Необходимо по заданным значениям n и k вычислить количество разреженных скобочных последовательностей из n символов, которые имеют глубину вложения скобок, равную k $(1 \le n \le 300,\ 0 \le k \le n)$. *Ответ записать по модулю* $(10^9 + 9)$.

Пусть $m[n,k]$ — количество разреженных правильных скобочных последовательностей-строк из n символов с глубиной вложения скобок, не превышающей k. Например, $m[3,1] = 4$, так как есть четыре разреженные скобочные последовательности «0()», «(0)», «()0» и «000». Тогда ответом на задачу будет значение выражения $m[n,k] - m[n,k-1]$. С помощью комбинаторных принципов сложения и умножения несложно выводится рекуррентное соотношение

$$m[n,k] = m[n-1,k] + \sum_{i=0}^{n-2} m[i,k-1] \cdot m[n-i-2,k],$$

при этом $m[i,0] = m[0,j] = 1$ для всех $i \ge 0$ и $j \ge 0$. Действительно, все рассматриваемые скобочные последовательности длины n разбиваются на два непересекающихся класса: строки, у которых первый символ «0», и строки, у которых этот символ — открывающаяся скобка «(». В первом случае количество таких строк, очевидно, равно $m[n-1,k]$. Скобочные последовательности второго класса начинаются с символа «(», которому соответствует некоторая закрывающая скобка «)». Между этими скобками и вне этих скобок получились две подстроки, состоящие соответственно из i и $n-i-2$ символов, где i принимает все возможные значения от 0 до $n-2$. Каждая подстрока является разреженной правильной скобочной последовательностью, у первой из них глубина вложения скобок не более $k-1$, а у второй — не более k. Общее число всех таких комбинаций равно $m[i,k-1] \cdot m[n-i-2,k]$. Суммируя по всем i от 0 до $n-2$, получим требуемую формулу для подсчета чисел $m[n,k]$.

Разреженная скобочная последовательность с глубиной вложения скобок k соответствует *пути Моцкина*, который не поднимается выше уровня k. В энциклопедии [4] количество путей Моцкина длины $n \ge 0$ и высоты $k \ge 0$ задаётся последовательностью A097862.

Микросхемы

[Открытый Кубок имени Е.В.Панкратьева, 2015, автор — *М. Киндер.*]

Наверное, вам известно, как читать микросхемы. Прежде всего, необходимо обратить внимание на особенности соединения контактов. На схемах контакты соединяются токопроводящими жилами. Если две жилы не пересекаются, то соединение между контактами, из которых выходят эти жилы, отсутствует. У вас в руках микросхема в виде круга, на границе которого расположено n контактов. Вам необходимо подсчитать количество способов нанести ровно k непересекающихся жил, каждая из которых соединяет ровно два контакта ($1 \le k \le n \le 40$).

Пусть $m[n,k]$ — количество способов нарисовать k непересекающихся линий с вершинами в n заданных точках окружности. Например, $m[4,1] = 6$, как это видно из рис. 1. Вообще говоря, $m[n,k]$ — это известная последовательность чисел Моцкина, которая встречается во многих комбинаторных задачах перечисления. В on-line энциклопедии [4] ненулевые числа $m[n,k]$ образуют последовательность [6], они совпадают также с количеством целочисленных путей длины n, у которых имеется ровно k шагов вверх [5]. С помощью комбинаторных принципов сложения и умножения несложно выводится рекуррентное соотношение

$$m[n,k] = m[n-1,k] + \sum_{i=0}^{n-2}\sum_{j=0}^{k-1} m[i,j]\cdot m[n-i-2,k-j-1],$$

при этом $m[i,0] = 1$ и $m[0,j] = 0$ для всех $i \ge 0$ и для всех $j \ge 1$. Приведенное рекуррентное соотношение теперь можно использовать для применения метода динамического программирования. Ограничения на переменные k и n выбраны с таким расчетом,чтобы в самом худшем случае процесс вычисления укладывался в отведенное время (как правило, это одна секунда на тест).

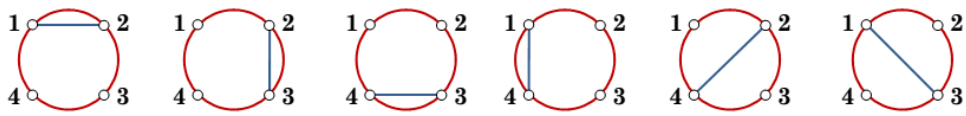

Рис. 1.

3. Заключение

Создание качественных олимпиадных задач по информатике является сложным и трудоемким процессом. Эта статья описывает рекурсивный подход к перечислению некоторых классов комбинаторных задач. Классические комбинаторные объекты — частые гости олимпиадных соревнований различного уровня. Комбинаторные проблемы, в которых они возникают, опираются на зависимость от рекуррентных соотношений и поэтому, чаще всего, решаются с помощью метода динамического программирования. Комбинаторика, конечно, не единственный раздел математики, который может дать интересные задачи для соревнований по программированию. Однако мы ориентируемся на него здесь ещё и потому, что многие комбинаторные задачи имеют забавные легенды и привлекательны для школьников и студентов.

Большинство задач и примеров в этой статье встречались на соревновании по спортивному программированию — Открытом кубке им. Е.В.Панкратьева (Гран-При Татарстан). Полные тексты всех этих задач доступны в Интернете [3]. Мы надеемся, что некоторые из приведенных задач позволят расширить область применения комбинаторики в соревнованиях по программированию различного уровня.

Литература

1. Burton B. Creating informatics olympiad tasks: exploring the black art / B.Burton, M.Hiron // Olympiads in Informatics.- 2008.- №9.-P.16-36.

2. Sloane N.J.A. On non-squashing partitions / N.J.A. Sloane, J.A. Sellers // Discrete Mathematics.- 2005.- №294(3), P.259-274.

3. Электронный ресурс удаленного доступа (http://www.icl.ru/turnir)

4. Sloane N.J.A. The On-Line Encyclopedia of Integer Sequences [http://oeis.org/]:this site is supported by donations to The OEIS Foundation.-1964.-Access mode:http://oeis.org/.

5. Sloane N.J.A. The On-Line Encyclopedia of Integer Sequences [http://oeis.org/]:this site is supported by donations to The OEIS Foundation/Triangular array of Motzkin polynomial coefficients.-1964.-Access mode:http://oeis.org/A055151.

6. Sloane N.J.A. The On-Line Encyclopedia of Integer Sequences [http://oeis.org/]:this site is supported by donations to The OEIS Foundation/Triangular array of ways of drawing k non-intersecting chords between n points on a circle; i.e. Motzkin polynomial coefficients.-1964.-Access mode: http://oeis.org/A080159

COMBINATORIAL OBJECTS OF ENUMERATION IN PROGRAMMING CONTESTS

M.I. Kinder

This paper describes a recursive approach to the enumeration of some classes of combinatorial tasks. Most tasks are used in the specific scope of teaching and learning informatics through olympiads and other competitions. Combinatorial problems can often lead to interesting and beautiful dynamic programming tasks, because they both depend on recurrence relations: formulae that solve a larger problem in terms of one or more smaller problems. Combinatorics of course is not the only branch of mathematics that can yield interesting tasks for programming contests. We focus on it here because many combinatorial problems have entertaining legends and they are easily accessible to students. Many of the examples in this paper are taken from the Open Cup named after E.V. Pankratiev (Grand-Prix of Tatarstan). Full texts for all of these problems are available on the Internet: www.icl.ru/turnir.

Keywords: programming contests, informatics olympiads, combinatorial tasks.