

ЗАДАЧИ ПЕРВОЙ ВСЕРОССИЙСКОЙ КОМАНДНОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ

Задача А РЕДАКТОР

Компания *Macrohard* выпустила новую версию своего редактора *Nottoobad*, который понимает некоторые голосовые команды. К сожалению, этих команд всего две – «повторить последнее слово» и «стереть последний символ». Причем, при исполнении команды «повторить последнее слово» редактор автоматически вставляет пробел, который разделяет слова.



Однако компания утверждает, что с помощью этого редактора можно набирать текст, нажимая клавиши на клавиатуре гораздо реже. Например, чтобы набрать фразу «this thin thing», достаточно нажать на клавиши на клавиатуре всего 6 раз (см. таблицу 1).

Чтобы повысить популярность своего продукта, компания решила провести конкурс, победителем которого станет тот, кто сможет набрать заданный набор слов в редакторе за наименьшее количество нажатий на клавиши. Причем первое сло-

во зафиксировано, а остальные могут быть набраны в произвольном порядке. То есть, если надо набрать слова «apple», «plum» и «argicote», то первым надо набрать «apple», а слова «plum» и «argicote» можно поменять местами.

Поскольку Вы собираетесь участвовать в конкурсе и у Вас есть знакомый в компании, который сообщил Вам по секрету набор слов, которые надо будет набрать, то неплохо бы написать программу, которая найдет порядок набора слов, при котором количество нажатий на клавиши будет минимальным.



Решение

Решение этой задачи напоминает известный алгоритм цифровой сортировки.

Заметим, что если некоторый набор слов начинается с одной и той же буквы, то их всегда выгодно набирать подряд. Действительно: предположим, что мы нашли оптимальное решение, в котором данное условие не выполняется. Предположим, к примеру, что слова, начинающиеся с буквы «а» набирались не подряд. Тог-

Действие	Нажатий	Содержимое документа
Набрать «this»	4	this
Сказать «повторить последнее слово»	0	this this
Сказать «стереть последний символ»	0	this thi
Набрать «n»	1	this thin
Сказать «повторить последнее слово»	0	this thin thin
Набрать «g»	1	this thin thing

Таблица 1

Блок	Описание	Кол-во нажатий
1	Различные слова	X1
2	Слова, начинающиеся с буквы «а»	X2
3	Различные слова, не начинающиеся с буквы «а»	X3
4	Слова, начинающиеся с буквы «а»	X4
5	Различные слова	X5

Таблица 2

Блок	Описание	Кол-во нажатий
1	Различные слова	X1
2	Слова, начинающиеся с буквы «а»	X2
4	Слова, начинающиеся с буквы «а»	Не более X4-1
3	Различные слова, не начинающиеся с буквы «а»	X3
5	Различные слова	Не более X5

Таблица 3

да порядок набора выглядит так, как в таблице 2. Переставим блоки, как в таблице 3.

Тем самым суммарное количество нажатий уменьшилось, по крайней мере, на 1. А это противоречит утверждению о том, что выбранный порядок нажатий был оптимальным.

Значит, сначала следует набрать все слова, которые начинаются на ту же букву, что и первое слово. Остальные слова следует разбить на группы, в которых все слова начинаются с одной и той же буквы, и набирать их по группам.

Заметим теперь, что порядок набора слов в каждой группе не влияет на количество нажатий, требуемое для набора остальных групп, поскольку при смене первой буквы слова для набора следующего в любом случае придется стереть все буквы предыдущего голосовыми командами (а всегда выгодно сказать «повторить последнее слово», поскольку в этом случае не приходится набирать пробел). Значит, порядок набора слов в каждой группе можно определять независимо.

Рассмотрим набор слов, начинающихся с одной буквы. Отбросив первую букву, получим задачу, эквивалентную ис-

ходной с тем лишь исключением, что вместо пробела слова теперь разделяет пробел и первая буква слов нашей группы. Значит, ее решение аналогично.

Осталось найти порядок слов, при котором слова, имеющие совпадающими первые i букв, не будут разделены словом, у которого одна из первых i букв отличается от букв данных слов. Ясно, что таким является, например, лексикографический (алфавит-

ный) порядок. Единственная проблема в том, что слово, заданное во входном файле первым, необходимо оставить первым. Можно предложить, например, такое ее решение: разделим слова на две группы – те, которые начинаются с той же буквы, что и первое слово, и все остальные. Как было показано выше, слова в этих группах можно упорядочивать независимо. Упорядочим слова во второй группе лексикографически, а в первой группе виртуально удалим у всех слов первую букву и повторим операцию для них. Зная порядок, в котором следует набирать слова, нетрудно подсчитать количество нажатий, которое для этого потребуются.

Задача В

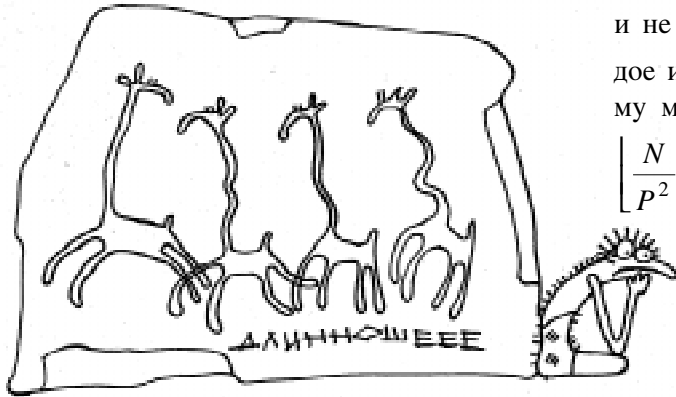
МАРСИАНСКИЕ ФАКТОРИАЛЫ

В 3141 году очередная экспедиция на Марс обнаружила в одной из пещер таинственные знаки. Они однозначно доказывали существование на Марсе разумных существ. Однако смысл этих таинственных знаков долгое время оставался неизвестным. Недавно один из ученых, профессор Очень-Умный, заметил один интересный факт: всего в надписях, составленных из этих знаков, встречается ровно K

различных символов. Более того, все надписи заканчиваются на длинную последовательность одних и тех же символов.

Вывод, который сделал из своих наблюдений профессор, потряс всех ученых Земли. Он предположил, что эти надписи являются записями факториалов различных натуральных чисел в системе счисления с основанием K . А символы в конце – это, конечно же, нули – ведь, как известно, факториалы больших чисел заканчиваются большим количеством нулей. Например, в нашей десятичной системе счисления факториалы заканчиваются на нули, начиная с $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$. А у числа $100!$ в конце следует 24 нуля в десятичной системе счисления и 48 нулей в системе счисления с основанием 6 – так что у предположения профессора есть разумные основания!

Теперь ученым срочно нужна программа, которая по заданным числам N и K найдет количество нулей в конце записи в системе счисления с основанием K числа $N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (N-1) \cdot N$, чтобы они могли проверить свою гипотезу. Вам придется написать им такую программу!



Решение

Для решения этой задачи достаточно заметить, что для того, чтобы число A заканчивалось на X нулей в системе счисления с основанием K , необходимо и достаточно, чтобы оно делилось без остатка на K^X . Действительно:

$$A = A_p K^p + A_{p-1} K^{p-1} + \dots + A_0 K^0.$$

Если оно заканчивается на X нулей, то

$$A_i = 0 \text{ для } i = 0, 1, \dots, X-1.$$

Значит,

$$A = A_p K^p + A_{p-1} K^{p-1} + \dots + A_x K^x = (A_p K^{p-x} + A_{p-1} K^{p-x-1} + \dots + A_x K^0) K^x$$

делится на K^x . Обратно, если $A = B \cdot K^x$, то представляя

$$B = B_Q K^Q + B_{Q-1} K^{Q-1} + \dots + B_0 K^0,$$

получаем, что в разложении A младшие X цифр будут нулями.

Значит, задача свелась к тому, чтобы определить, на какую максимальную степень K делится число $N!$. Поскольку $N!$ может быть очень велико, непосредственное его вычисление с целью такой проверки невозможно. Разложим число K на простые множители. Пусть $K = P_1^{a_1} P_2^{a_2} \dots P_s^{a_s}$, тогда, если $N!$ делится на соответственно $P_i^{b_i}$, то $N!$ делится на K^Z , где

$$Z = \min_{i=1..s} \left\{ \frac{b_i}{a_i} \right\} \text{ (см. примечание } ^1).$$

Единственная оставшаяся проблема – как для простого числа P найти максимальную его степень, на которую делится $N!$.

Для этого применим следующие соображения: количество чисел, кратных P и не превышающих N , равно $\left\lfloor \frac{N}{P} \right\rfloor$. Каждое из этих чисел даст по одному простому множителю P в $N!$. Но, кроме того,

$\left\lfloor \frac{N}{P^2} \right\rfloor$ чисел дадут еще по одному P ,

$\left\lfloor \frac{N}{P^3} \right\rfloor$ – еще по одному и т. д. Значит,

$$b_i = \left\lfloor \frac{N}{P_i} \right\rfloor + \left\lfloor \frac{N}{P_i^2} \right\rfloor + \left\lfloor \frac{N}{P_i^3} \right\rfloor + \dots$$

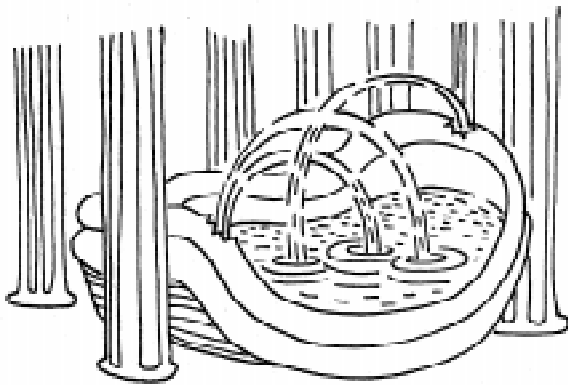
Заметим, что суммирование можно остановить, когда очередное слагаемое равно 0.

Задача С ФОНТАН

Администрация одного института решила построить в холле фонтан. По плану администрации, фонтан должен иметь форму круга с максимально возмож-

ным радиусом. Дизайнеру сообщили, что холл института имеет вид прямоугольника размером $X \times Y$ метров. Однако, когда дизайнер стал выбирать место для фонтана, он столкнулся с серьезной проблемой: в холле института обнаружилось N круглых колонн, снести которые не представляется возможным.

Таким образом, у него появилась проблема: где следует поместить фонтан, чтобы он имел максимально возможный радиус и не имел ненулевого по площади пересечения с колоннами. Вам предстоит помочь ему в решении этой нелегкой задачи.



Решение

Это самая сложная задача из предлагавшихся на олимпиаде. Идея ее решения следующая: предположим, мы хотим построить фонтан радиуса R . Тогда, чтобы проверить, что мы можем это сделать, увеличим радиус всех колонн на R , а каждую сторону холла уменьшим на $2R$ (так, чтобы его центр оставался на том же месте). Тогда, чтобы проверить, можем ли мы построить такой фонтан, достаточно выяснить, покроют ли после этой операции круги колонн прямоугольник холла полностью. Если это так, то построить фонтан данного радиуса, очевидно, не удастся.

Чтобы проверить, что множество кругов покрывает заданный прямоугольник, можно использовать следующий алгоритм: достаточно проверить, что точки пересечения любых двух окружностей, лежащие внутри прямоугольника, точки пересече-

ния окружностей с прямоугольником и углы прямоугольника покрываются каким-либо другим кругом (область, не покрываемая кругами, всегда содержит одну из таких точек).

Для нахождения точек пересечения двух окружностей необходимо решить следующую систему уравнений относительно x и y :

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = r_2^2 \end{cases}$$

После замены $x - x_1 = \bar{x}$, $y - y_1 = \bar{y}$ и обозначения $x_1 - x_2 = \Delta x$, $y_1 - y_2 = \Delta y$, система принимает вид:

$$\begin{cases} \bar{x}^2 + \bar{y}^2 = r_1^2 \\ \bar{x}^2 + 2\bar{x}\Delta x + \Delta x^2 + \bar{y}^2 + 2\bar{y}\Delta y + \Delta y^2 = r_2^2 \end{cases}$$

После вычитания первого уравнения из второго, получаем линейную зависимость между \bar{x} и \bar{y} :

$$\begin{cases} \bar{x}^2 + \bar{y}^2 = r_1^2 \\ 2\bar{x}\Delta x + \Delta x^2 + 2\bar{y}\Delta y + \Delta y^2 = r_2^2 - r_1^2 \end{cases}$$

Отсюда несложно получить квадратное уравнение для \bar{x} или \bar{y} .

Пересечение прямой с окружностью выполняется аналогично – там линейная зависимость сразу задается принадлежностью к прямой.

Затем для проверки того, что точка лежит внутри круга, достаточно проверить, что расстояние от точки до центра круга меньше его радиуса.

Однако этот метод сопряжен с техническими трудностями, связанными с проблемами, которые возникают при разрешении линейной зависимости относительно одной из переменных – при этом возникает опасность деления на 0, что требует рассмотрения двух различных случаев. Имеется еще один способ проверки того, что набор кругов с заданной точностью покрывает прямоугольник, правда, существенно использующий факт наличия требуемой точности вычислений. Предположим, что мы хотим проверить, покрывается ли некоторый прямоугольник за-

данным множеством кругов. Если все четыре его вершины покрываются одним кругом, то, очевидно, прямоугольник полностью покрывается кругами. В противном случае, разобьем прямоугольник на четыре одинаковых прямоугольника и рекурсивно проверим, что они покрываются кругами. Если в процессе рекурсии сторона прямоугольника стала меньше некоторой заданной величины, можно подозревать, что этот прямоугольник полностью кругами не покрывается.

Задача D ДЕТСКИЙ ПРАЗДНИК

Организаторы детского праздника планируют надуть для него M воздушных шариков. С этой целью они пригласили N добровольных помощников, i -й среди которых надувает шарик за T_i минут, однако каждый раз после надувания Z_i шариков устает и отдыхает Y_i минут. Теперь организаторы праздника хотят узнать, через какое время будут надуты все шарики при наиболее оптимальной работе помощников и сколько шариков надует каждый из них. (Если помощник надул шарик и должен отдохнуть, но больше шариков ему надувать не придется, то считается, что он закончил работу сразу после окончания надувания последнего шарика, а не после отдыха).

Решение

Для решения этой задачи можно применить жадный алгоритм – будем добавлять шарики по очереди и давать шарик для надувания тому помощнику, который закончит этот процесс раньше других.



Задача E СИМПАТИЧНЫЕ УЗОРЫ

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $M \times N$ метров.

Однако, при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во-первых, каждый новый клиент, очевидно, захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во-вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета. На рисунке 1а показаны примеры различных симпатичных узоров, а на рисунке 1б – несимпатичных.

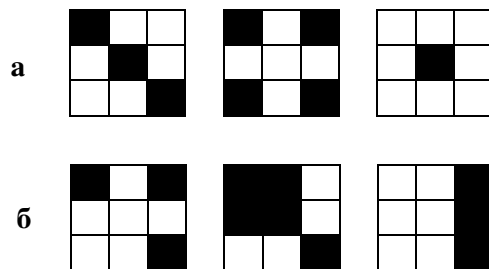
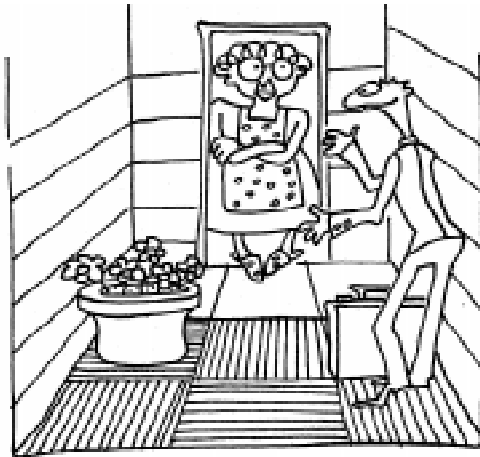


Рисунок 1

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Решение

Для решения этой задачи применим динамическое программирование. Сна-



чала заметим, что общее количество узоров не превышает $2^{M \cdot N} \leq 2^{30}$, что означает, что ответ на задачу поместится в типы данных `longint` в Паскале и `long` в Си. Будем также, не ограничивая общности, считать, что $M \leq N$, а значит, $M \leq 5$.

Рассмотрим узор размера $M \times K$, где K может меняться в пределах от 1 до N , а M зафиксировано. Назовем профилем узора двоичное число, кодирующее его самый правый столбец, где черному квадратику отвечает 1, а белому – 0.

На рисунке 2 приведен пример узора 3×3 с профилем 001. Заметим, что количество различных профилей не зависит от K и равно 2^M . Значит, поскольку $M \leq 5$, количество различных профилей не превышает 32.

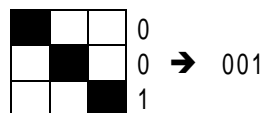


Рисунок 2

Теперь для каждого K и каждого профиля Z посчитаем количество $A[K][Z]$ симпатичных узоров размера $M \times K$, имеющих профиль Z . Поскольку в узоре $M \times 1$ одноцветных квадратов 2×2 встретиться не может, для всех профилей $A[1][Z]=1$. Далее, при переходе от K к $K+1$ следует для данного профиля Z сложить те значения $A[K][Y]$, где в Y хог Z не встречается подряд двух нулей (это отвечает нарушению симпатичности).

Наконец, ответ на задачу получается при сложении по всем Z значений $A[N][Z]$.

Задача F КУБИКИ

Родители подарили Пете набор детских кубиков. Поскольку Петя скоро пойдет в школу, они купили ему кубики с буквами. На каждой из шести граней каждого кубика написана буква.

Теперь Петя хочет похвастаться перед старшей сестрой, что научился читать. Для этого он хочет сложить из кубиков ее имя. Но это оказалось довольно сложно сделать – ведь разные буквы могут находиться на одном и том же кубике, и тогда Петя не сможет использовать обе буквы в слове. Правда, одна и та же буква может встречаться на разных кубиках. Помогите Пете!

Дан набор кубиков и имя сестры. Выясните, можно ли выложить ее имя с помощью этих кубиков, и если да, то в каком порядке следует выложить кубики.

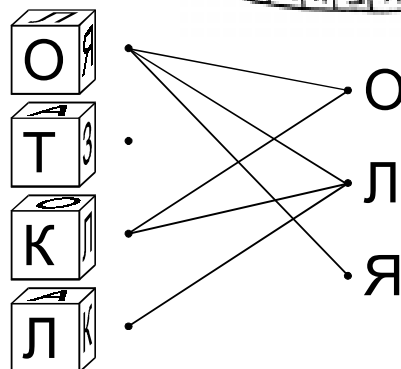
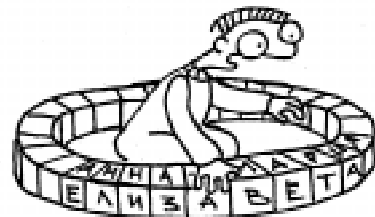


Рисунок 3

Решение

Эта задача сводится к нахождению максимального паросочетания в двудольном графе.² Рассмотрим кубики и буквы имени сестры как вершины графа и со-

единим кубик ребром с буквой, если эта буква написана на этом кубике (рисунок 3). Заметим, что граф действительно двудольный, а выбор кубиков для выкладки имени эквивалентен построению паросочетания. Поскольку количество ребер в паросочетании не превышает количества вершин в меньшей доле, то искомое паросочетание действительно максимально.³

Задача G ВОЛШЕБНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

Недавно Петя научился считать. Он тут же заметил, что число 2 обладает замечательным свойством – $2 + 2 = 2 \times 2$. Его старший брат Ваня тут же объяснил ему, что дело не в двойке.

«Дело в том, что последовательность 2, 2 – волшебная, – сказал Пете Ваня. – Волшебная последовательность – это такая последовательность натуральных чисел, что сумма ее членов равна их произведению. Например, последовательность 1, 2, 3 – тоже волшебная.»

Петя тут же сложил 1, 2 и 3, потом перемножил их и обрадовался.

Теперь Петя хочет найти более длинные волшебные последовательности. Помогите ему!

Решение

Это – несложная задача на творческий поиск. После попыток найти последовательности для различных небольших N , нетрудно заметить, что последовательность $\underbrace{1, 1, \dots, 1}_{N-2}, 2, N$ удовлетворяет условию задачи.



Задача H ПОХОЖИЕ МАТРИЦЫ

Рассмотрим таблицу, состоящую из N строк и M столбцов. Если в каждой ячейке такой таблицы стоит целое число, назовем такую таблицу *целочисленной матрицей*. Скажем, что эта матрица *кратна числу p* , если все числа в ее ячейках кратны p .

Рассмотрим теперь суммы элементов матрицы по строкам и столбцам, соответственно. Обозначим сумму чисел i -й строки через H_i , а сумму чисел j -го столбца через V_j . Упорядоченный набор чисел $(H_1, H_2, \dots, H_N, V_1, V_2, \dots, V_M)$ назовем *профилем матрицы*. Скажем, что матрица *почти кратна p* , если все числа, входящие в ее профиль, кратны p .

Почти кратная 5 матрица и ее профиль изображены на рисунке 4.

6	7	2	15
2	2	31	35
7	1	7	15
			15 10 40

Рисунок 4

Если две матрицы A и B имеют одинаковый размер, причем элемент, стоящий на пересечении i -й строки и j -го столбца в матрице A , отличается от соответствующего элемента матрицы B не более чем на p , скажем, что A отличается от B не более чем на p . Скажем, что матрица B *похожа* на матрицу A относительно числа p , если

1. B отличается от A не более чем на p .
2. Профили B и A совпадают.

На рисунке 5 изображены две похожие относительно числа 5 матрицы, первая из них почти кратна 5, а вторая кратна 5. Третья матрица на рисунке 2 тоже кратна 5, но не похожа на первую (хотя похожа на вторую).

Дано число p и почти кратная p матрица A . Ваша задача – найти такую матрицу B , чтобы она была кратна p и похожа на A относительно p .

А		
6	7	2
2	2	31
7	1	7

почти кратна 5

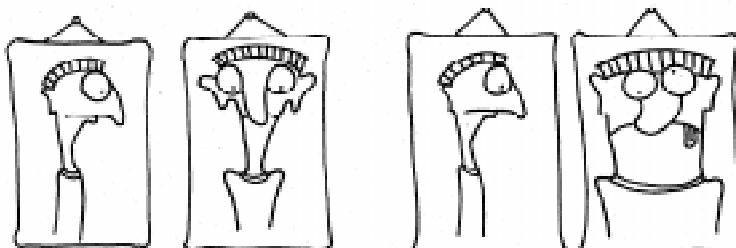
В		
5	5	5
0	5	30
10	0	5

кратна 5,
похожа на А

С		
5	5	5
0	0	35
10	5	0

кратна 5, но
непохожа на А

Рисунок 5



Решение

Будем итеративно улучшать матрицу, следя за тем, чтобы количество чисел, не кратных p уменьшалось и матрица не становилась непохожей на исходную.

Найдем в матрице (двигаясь слева направо и сверху-вниз)⁴ первую ячейку, число в которой не кратно p . Пусть это ячейка (i, j) . Поскольку сумма чисел в этой строке кратна p , то в ней найдется еще одно число, не кратно p , скажем в ячейке (i, k) . Уменьшим наше число

6	7	2
2	2	31
7	1	7

→

5	8	2
2	2	31
7	1	7

Рисунок 6

6	7	2
2	2	31
7	1	7

→

5	8	2
2	2	31
7	1	7

→

5	8	2
2	1	31
7	1	7

→

5	8	2
2	1	32
7	1	7

→

5	8	2
2	1	32
7	1	6

→

5	8	2
2	1	32
8	1	6

Рисунок 7

на 1, а второе число увеличим на 1 (рисунок 6).

Сумма чисел в строке не изменилась, правда, испортились суммы в столбцах j и k . Рассмотрим столбец k . Поскольку в нем было число, не кратное p , а сумма чисел в нем кратна p , то в нем найдется еще одно число, не кратное p , скажем, в ячейке (l, k) . Уменьшим его на единицу и повторим рассуждения для строки l . Рано или поздно мы снова попадем в столбец j , тогда увеличив соответствующее число на 1, мы получим, что профиль матрицы не изменится, первое не кратное p число уменьшится на 1, причем ни одно число не будет отличаться от своего исходного значения более чем на p , поскольку все изменения за один шаг делаются ровно на 1, значит, чтобы измениться более чем на p , числу попутно нужно принять значение, кратное p , а после этого оно уже не будет меняться.

Осталось показать, что, если повторять эту операцию, рано или поздно мы получим матрицу, кратную p . Но, действительно, если первое не кратное p число в матрице имело остаток x от деления на p , то в течение x итераций оно будет оставаться первым не кратным p числом в матрице, после x итераций станет кратным p . Значит, не более, чем через $m \cdot n \cdot p$ итераций матрица станет кратна p (рисунок 7).

Полные решения задач (программы) находятся на дискете к журналу.

¹ $\lfloor a \rfloor$ – ближайшее целое, не превышающее a .

² Граф называется двудольным, если его вершины можно разбить на два множества (Л и П) так, что любое ребро графа соединяет вершину из «левого» множества Л с вершиной из «правого»

множества P . Паросочетанием называется любое подмножество попарно не смежных ребер двудольного графа.

³ Для нахождения максимального паросочетания разработаны стандартные алгоритмы. Наиболее популярный алгоритм, использующий удлиняющиеся чередующиеся цепи, описан, например, в книге А.В. Ахо, Дж.Э. Хопкрофта, Дж.Д. Ульмана «Структуры данных и алгоритмы», нахождение максимального паросочетания через поиск максимального потока в сети с помощью алгоритма Форда-Фалкерсона описано, например, в книге Т. Кормена, Ч. Лейзерсона, Р. Ривеста «Алгоритмы: построение и анализ», теоретическое исследование проблемы максимального паросочетания можно найти, например, в книге И.В. Романовского «Дискретный анализ».

⁴ Более формально: будем обозначать ячейку на пересечении i -й строки и j -го столбца (i, j) . Скажем, что ячейка (i, j) идет раньше (k, l) в матрице, если $i < k$ или $i = k, j < l$ (лексикографический порядок). Скажем, что ячейка (i, j) , удовлетворяющая некоторому свойству, – первая, удовлетворяющая этому свойству, если для всех остальных ячеек (k, l) , удовлетворяющих этому свойству, (i, j) идет раньше (k, l) в матрице. Несложно показать, что наше определение корректно, то есть, если некоторому свойству удовлетворяет хотя бы одна ячейка, то найдется первая ячейка в матрице, удовлетворяющая этому свойству.

***Парфенов Владимир Глебович,
профессор кафедры компьютерных
технологий СПбГИТМО (ТУ),
руководитель отделения
прикладной математики, физики
и компьютерных технологий.***

***Станкевич Андрей Сергеевич,
студент СПбГИТМО (ТУ),
председатель жюри первой
Всероссийской командной
олимпиады школьников
по информатике.***

НАШИ АВТОРЫ