МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
БОРИСОГЛЕБСКИЙ ФИЛИАЛ
(БФ ФГБОУ ВО «ВГУ»)

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
Практикум по решению задач повышенной сложности по информатике

1. Код и наименование направления подготовки:

44.03.01 Педагогическое образование

2. Профиль подготовки:

Информатика и информационные технологии в образовании

3. Квалификация (степень) выпускника:

Бакалавр

4. Форма обучения:

Заочная

5. Кафедра, отвечающая за реализацию дисциплины:

Кафедра прикладной математики, информатики, физики и методики их преподавания

6. Составитель:

Е.А. Позднова, кандидат педагогических наук, доцент

7. Методические указания для обучающихся по освоению дисциплины

Приступая к изучению учебной дисциплины, целесообразно ознакомиться с учебной программой дисциплины, электронный вариант которой размещён на сайте БФ ВГУ.

Это позволит обучающимся получить четкое представление о:

- перечне и содержании компетенций, на формирование которых направлена дисциплина;
- основных целях и задачах дисциплины;
- планируемых результатах, представленных в виде знаний, умений и навыков, которые должны быть сформированы в процессе изучения дисциплины;
- количестве часов, предусмотренных учебным планом на изучение дисциплины, форму промежуточной аттестации;
- количестве часов, отведенных на аудиторные занятия и на самостоятельную работу;
- системе оценивания учебных достижений;
- учебно-методическом и информационном обеспечении дисциплины.

Знание основных положений, отраженных в рабочей программе дисциплины, поможет обучающимся ориентироваться в изучаемом курсе, осознавать место и роль изучаемой дисциплины в подготовке выпускника, строить свою работу в соответствии с требованиями, заложенными в программе.

Основными формами аудиторных занятий по дисциплине являются практические и лабораторные занятия.

В ходе подготовки к лабораторным занятиям необходимо изучить в соответствии с вопросами для повторения основную литературу, ознакомиться с дополнительной литературой. Кроме того, следует ответить на контрольные вопросы, изучить образцы решения задач.

При подготовке к промежуточной аттестации необходимо выполнить практические индивидуальные задания и подготовить отчет по лабораторным работам. Рекомендуется источники, перечисленные в списке литературы в рабочей программе дисциплины, а также ресурсы электронно-библиотечных систем.

.

8. Методические материалы для обучающихся по подготовке к практическим и лабораторным занятиям

№ п/п	Темы практических занятий	Содержание практических занятий
1	Информация и информационные процессы	Понятие информации. Свойства информации. Информационные процессы Единицы измерения информации. Содержательный подход к измерению информации. Алфавитный подход к измерению информации Кодирование информации Системы счисления Двоичное кодирование текстовой информации. Аналоговый и дискретный способы представления изображений и звука Двоичное кодирование графической информации.

		Двоичное кодирование звуковой информации.
2	Моделирование и формализация	Понятие модели Материальные и информационные модели Этапы моделирования Понятие формализации
3	Математические и логические основы информатики	Основные понятия математической логики. Высказывания, логические операции, кванторы, истинность высказывания. Основные законы алгебры логики. Преобразование логических выражений. Построение таблиц истинности. Решение логических задач. Цепочки (конечные последовательности), деревья, списки, графы.
4	Элементы теории алгоритмов	Формализация понятия алгоритма. Построение алгоритмов для конкретного исполнителя с фиксированным набором команд, практические вычисления. Цепочки (конечные последовательности), матрицы (массивы), псевдослучайные последовательности. Сортировка. Выигрышные стратегии.
5	Языки программирования	Основные конструкции языка программирования. Система программирования. Основные этапы разработки программ. Реализация основных операций работы с массивами на языке программирования. Разбиение задачи на подзадачи.

9. Задания к практическим занятиям

Тема «Информация и информационные процессы»

1. *При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 11 символов. Из соображений информационной безопасности каждый пароль должен содержать хотя бы 2 десятичных цифры, как прописные, так и строчные латинские буквы, а также не менее 2-х символов из 6-символьного набора: «&», «#», «$», «*», «!», «@». В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно*

пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 30 пользователях потребовалось 900 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Решение:

1) если бы мы знали точно, сколько цифр и сколько специальных символов содержит пароль и где точно они расположены, можно было бы использовать «раздельное» кодирование: на кодирование цифр использовать по 4 бита (2^4 > 10), на кодирование спецсимволов – по 3 бита (2^3 > 6), а на кодирование остальных символов (латинских букв) – по 6 бит (2^6 > 26·2=52)

2) поскольку количество и месторасположение цифр и спецсимволов а пароле неизвестно, нужно рассматривать полный набор символов: $10 + 6 + 26 \cdot 2 = 68$

3) при этом на каждый символ нужно выделить 7 бит ($2^7 > 68$)

4) на 11 символов пароля выделяется 77 бит, округляя вверх до целого числа байт получаем 10 байт (80 бит) на пароль

5) на одного пользователя выделяется 900 : 30 = 30 байт

6) на дополнительную информацию остается 30 – 10 = 20 байт

ответ: 20.

2. По каналу связи с помощью равномерного двоичного кода передаются сообщения, содержащие только 4 буквы: X, Y, Z, W; для кодировки букв используются кодовые слова длины 5. При этом для набора кодовых слов выполнено такое свойство: любые два слова из набора отличаются не менее чем в трёх позициях. Это свойство важно для расшифровки сообщений при наличии помех. Для кодирования букв X, Y, Z используются 5-битовые кодовые слова:

X: 01111, Y: 00001, Z: 11000. Определите 5-битовое кодовое слово для буквы W, если известно, что оно начинается с 1 и заканчивается 0.

Решение:

1) По условию кодовое слово для буквы W соответствует маске 1***0, где вместо звёздочек можно поставить 0 или 1.

2) Найдем количество позиций, в которых отличается это кодовое слово от известных кодовых слов букв X, Y и Z (расстояние Хэмминга):

X: 01111 Y: 00001 Z: 11000
W: 1***0 W: 1***0 W: 1***0
2+? 2+? 0+?

Знаки вопроса обозначают неизвестные неотрицательные числа – количество различающихся позиций в тех битах, которые в кодовом слове для буквы W неизвестны.

3) Как видим, наиболее критичная ситуация сложилась для пары Z-W. Для того, чтобы эти кодовые слова различались в трёх позициях, все неизвестные биты кодового слова буквы W должны иметь значения, обратные соответствующим битам кодового слова для буквы Z, то есть, W = 10110

4) Проверяем полученное кодовое слово: находим расстояние Хэмминга в парах X-W и Y-W:

X: 01111 Y: 00001 Z: 11000
W: 10110 W: 10110 W: 10110
3 4 3

5) Как видим, для все пар расстояние не меньше трёх, что соответствует условию задачи.

Ответ: 10110.

3. *Вася составляет 3-буквенные слова, в которых есть только буквы В, Е, С, Н , А, причём буква А используется в каждом слове хотя бы 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?*

Решение

1) буква А может стоять на одном из трёх мест: А**, *А*, **А, где * обозначает любой из пяти символов

2) в каждом случае в остальных двух позициях может быть любая из пяти букв

3) для шаблона А** получаем (перемножая количество вариантов для каждой позиции)

$1 \cdot 5 \cdot 5 = 25$ слов

4) для шаблона *А* тоже получим 25 слов, но нужно учесть, что все слова, в который первая буква А мы уже подсчитали, поэтому считаем только слова, где на первом место стоит какая-то другая буква (В, Е, С или Н)

5) отсюда находим, что шаблон *А* добавляет $4 \cdot 1 \cdot 5 = 20$ новых слов

6) рассматривая шаблон **А, не учитываем уже подсчитанные слова, в которых буква А есть на первом или втором местах, количество новых слов – $4 \cdot 4 \cdot 1 = 16$

7) всего получается $25 + 20 + 16 = 61$ слово

8) Ответ: 61.

4. *Вася составляет 5-буквенные слова, в которых есть только буквы С, Л, О, Н, причём буква С используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?*

Решение

1) буква С может стоять на одном из пяти мест: С****, *С***, **С**, ***С* и ****С, где * обозначает любой из оставшихся трёх символов

2) в каждом случае в остальных четырёх позициях может быть любая из трёх букв Л, О, Н, поэтому при заданном расположении буквы С имеем $3^4 = 81$ вариант

3) всего вариантов $5 \cdot 81 = 405$.

4) Ответ: 405.

5. *Сколько существует различных символьных последовательностей длины 5 в четырёхбуквенном алфавите {A, C, G, T}, которые содержат ровно две буквы А?*

Решение

1) рассмотрим различные варианты слов из 5 букв, которые содержат две буквы А и начинаются с А:

АА***   А*А**   А**А*   А***А

Здесь звёздочка обозначает любой символ из набора {C, G, T}, то есть один из трёх символов.

2) итак, в каждом шаблоне есть 3 позиции, каждую из которых можно заполнить тремя способами, поэтому общее число комбинаций (для каждого шаблона!) равно $3^3 = 27$

3) всего 4 шаблона, они дают $4 \cdot 27 = 108$ комбинаций

4) теперь рассматриваем шаблоны, где первая по счёту буква А стоит на второй позиции, их всего три:

*АА**   *А*А*   *А**А

они дают $3 \cdot 27 = 81$ комбинацию

5) два шаблона, где первая по счёту буква А стоит на третьей позиции:

**АА*   **А*А

они дают $2 \cdot 27 = 54$ комбинации

6) и один шаблон, где сочетание АА стоит в конце

***АА

они дают 27 комбинаций

7) всего получаем (4 + 3 + 2 + 1) · 27 = 270 комбинаций

8) ответ: 270.

6. *Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 30 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б; пропускная способность канала связи с городом Б в 4 раза выше, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б? В ответе запишите только целое число, единицу измерения писать не нужно.*

Решение

1) объём музыкального файла вычисляется по формуле $I = f \cdot r \cdot k \cdot t$, где *f* – частота дискретизации, *r* – разрешение (глубина кодирования), *k* – количество каналов, *t* – время звучания

2) при повышении разрешения (количества битов на хранения одного отсчёта) в 2 раза объём файла (при прочих равных условиях) увеличивается в 2 раза, поэтому время тоже увеличится в 2 раза

3) при снижении частоты дискретизации (количества хранимых отсчётов за 1 секунду) в 1,5 раза объём файла (при прочих равных условиях) уменьшается в 1,5 раза, поэтому время тоже уменьшится в 1,5 раза

4) при увеличении пропускной способности канала связи (здесь это то же самое, что и скорость передачи данных) в 4 раза время передачи (при прочих равных условиях) уменьшится в 4 раза

5) поэтому исходное время передачи файла нужно

а) умножить на 2

б) разделить на 1,5

в) разделить на 4

6) получается 30 · 2 / 1,5 / 4 = 10 секунд

Ответ: 10.

Тема «Моделирование и формализация»

1. Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A		4				
B	4		6	3	6	
C		6			4	
D		3			2	
E		6	4	2		5
F					5	

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

Решение. Варианты маршрутов:

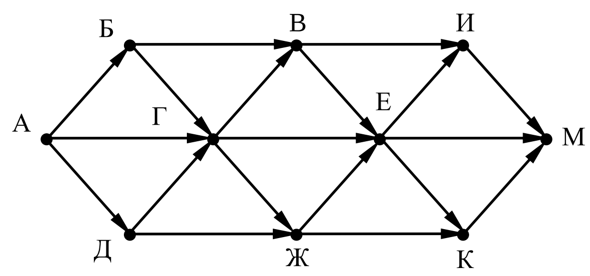
A-B-C-E-F. Длина маршрута 4 + 6 + 4 + 5 = 19
A-B-D-E-F. Длина маршрута 4 + 3 + 2 + 5 = 14
A-B-E-F. Длина маршрута 4 + 6 + 5 = 15
Видно, что кратчайший путь равен 14.

2. Между населёнными пунктами A, B, C, D, E, F, G построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице значает, что прямой дороги между пунктами нет.

	A	B	C	D	E	F
A		2	4	8		16
B	2			3		
C	4			3		
D	8	3	3		5	3
E				5		5
F	16			3	5	

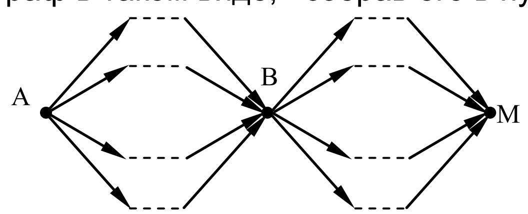
Определите длину кратчайшего пути между пунктами A и F, проходящего через пункт E. Передвигаться можно только по указанным дорогам.

3 *На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей, ведущих из города А в город М и проходящих через город В?*

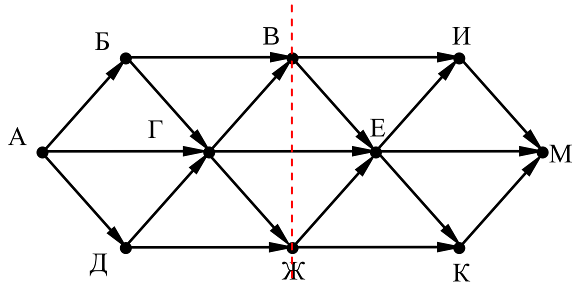


Решение:

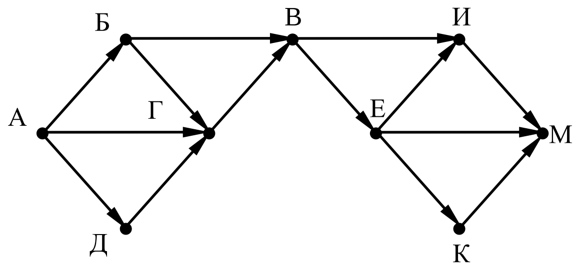
9) для того, чтобы оставить только маршруты, проходящие через вершину В, нужно представить граф в таком виде, «собрав его в пучок» около вершины В:



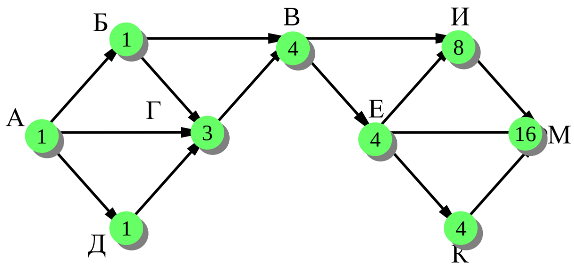
10) проведём сечение графа через вершину В:



11) обратим внимание на такой факт: если мы перешли через линию сечения из левой части в правую по ребру ГЕ или через вершину Ж, мы уже никак не попадём в вершину В (нет рёбер с «обратным направлением», поэтому эти маршруты запрещены; для более сложных случаев, когда такие рёбра с «обратным направлением» есть, нужно перерисовать граф (или провести сечение иначе) так, чтобы все вершины, ИЗ которых можно попасть в В, оказались слева от линии сечения
12) в данном случае выбрасывается вершина Ж, все связанные с ней рёбра, и ребро ГЕ:

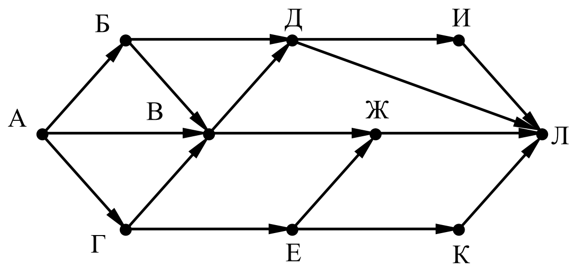


13) дальше используем стандартный метод (см. разбор следующей задачи)
14) покажем только окончательный результат:



15) Ответ: 16.

4 *На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л?*



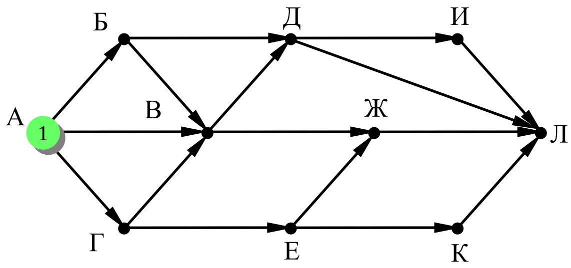
Решение:

1) будем обозначать через N_X количество различных путей из города А в город Х
2) для города А есть только один маршрут – никуда не двигаться, поэтому $N_A = 1$
3) для любого города Х количество маршрутов N_X можно вычислить как

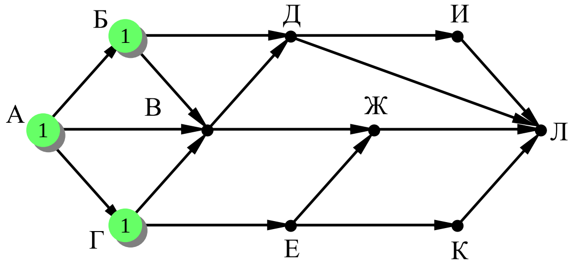
   $N_x = N_y + \ldots + N_z$

   где сумма взята по всем вершинам, из которых есть прямой путь в вершину Х; например,

   $N_Л = N_И + N_Ж + N_К$
4) около каждого города будем записывать количество маршрутов из А в этот город
5) начнем считать количество путей с начала маршрута – с города А:



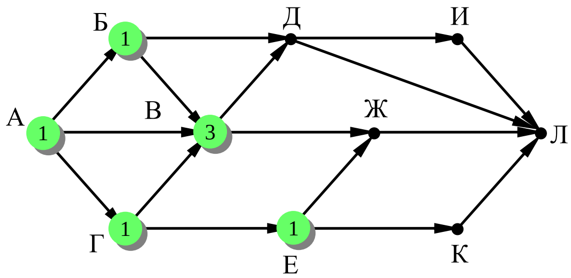
6) теперь находим те вершины, в которые можно попасть напрямую из уже рассмотренных вершин (пока – только из А), это Б и Г, для них тоже количество путей равно 1:



7) теперь можно определить количество путей для В и Е; в В можно приехать только из А, Б и Г, а в Е – только из Г:

$N_В = N_А + N_Б + N_Г = 1 + 1 + 1 = 3$

$N_Е = N_Г = 1$

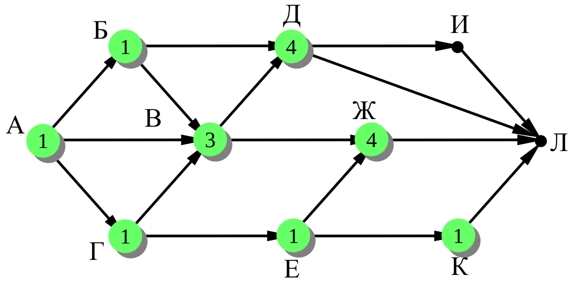


8) теперь можно определить количество путей для Д, Ж и К; в Д можно приехать только из Б и В, в Ж – из В и Е, а в Е – только из Г:

$N_Д = N_Б + N_В = 1 + 3 = 4$

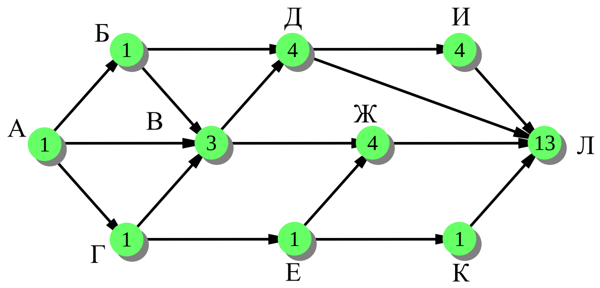
$N_Ж = N_В + N_Е = 3 + 1 = 4$

$N_К = N_Е = 1$



9) теперь можно определить количество путей для И, куда можно приехать только из Д ($N_И = N_Д$) и, наконец, для Л:

$N_Л = N_Д + N_И + N_Ж + N_К = 13$



10) Ответ: 13.

Тема «Математические и логические основы информатики»

1. Для какого наибольшего целого числа А формула $((x \le 9) \to (x \cdot x \le A)) \wedge ((y \cdot y \le A) \to (y \le 9))$ *тождественно истинна (то есть принимает значение 1 при любых целых неотрицательных значениях переменных x и y)?*

Решение:

заметим, что здесь два условия, которые объёдиняются с помощью логической операции «И»:

$(x \le 9) \to (x \cdot x \le A)$

$(y \cdot y \le A) \to (y \le 9)$

Необходимо, чтобы оба условия были выполнены одновременно; к счастью, первое зависит только от переменной *x*, а второе – только от переменной *y*, поэтому их можно рассматривать отдельно: каждое из них задает некоторое ограничение на значение *A* рассмотрим первое условие: $(x \le 9) \to (x \cdot x \le A)$. Для того чтобы импликация была истинной, нужно не допустить варианта $1 \to 0$, то есть при истинной левой части правая часть тоже должна быть истинной.

Это значит, что для всех $0 < x \le 9$ мы должны обеспечить $x \cdot x \le A$, то есть выбрать $A \ge x \cdot x$ для все допустимых значений *x*. Очевидно, что для этого необходимо и достаточно выбрать $A \ge 9 \cdot 9 = 81$. Таким образом, мы определили минимальное допустимое значение $A = 81$.

Теперь рассмотрим второе условие: $(y \cdot y \le A) \to (y \le 9)$. Чтобы оно было истинно, нужно не допустить варианта $1 \to 0$. Выбором А мы можем влиять на левую часть, но не на правую. «Угрозу» представляет вариант, когда правая часть ложна, то есть $y > 9$. В этом случае нам нужно сделать левую часть ложной, то есть обеспечить выполнение условия $y \cdot y > A$.

Для выбора максимального *A* возьмем минимальное значение *y*, для которого $y > 9$. Это даёт условие $10 \cdot 10 > A$, откуда следует $A < 100$

Таким образом, максимально допустимое значение *A* равно 99.

Ответ: 99.

2. *Введём выражение M & K, обозначающее поразрядную конъюнкцию M и K (логическое «И» между соответствующими битами двоичной записи). Определите наименьшее натуральное число a, такое что выражение*

(*x* & 125 ¹ 1) ∨ ((*x* & 34 = 2) → (*x* & *a* = 0))

тождественно истинно (то есть принимает значение 1 при любом натуральном значении переменной x)?

Решение:

1) используя результаты теоретической части, перепишем выражение в виде $\overline{Z}_{124} + Z_1 + (Z_{32} \cdot \overline{Z}_2 \to A) = 1$

где Z_{124} **=** (x & 124 = 0), Z_1 **=** (x & 1 = 0), Z_2 **=** (x & 2 = 0), **A** = (x & a = 0)

2) раскроем импликацию по формуле $A \to B = \overline{A} + B$:
   $\overline{Z}_{124} + Z_1 + \overline{Z_{32} \cdot \overline{Z}_2} + A = 1$
3) применим закон де Моргана $\overline{A \cdot B} = \overline{A} + \overline{B}$:
   $\overline{Z}_{124} + Z_1 + \overline{Z}_{32} + Z_2 + A = 1$
4) перейдём к импликации, в которой нет выражений с инверсиями (операциями «НЕ»):
   $(\overline{Z}_{124} + \overline{Z}_{32}) + Z_1 + Z_2 + A = 1$
   $(\overline{Z_{124} \cdot Z_{32}}) + Z_1 + Z_2 + A = 1$
   $(Z_{124} \cdot Z_{32}) \to (Z_1 + Z_2 + A) = 1$
5) преобразуем левую часть выражения:
   $Z_{124} \cdot Z_{32} = Z_{124\,\mathrm{or}\,32} = Z_{124}$
   так что $Z_{124} \to (Z_1 + Z_2 + A) = 1$
6) используя свойство импликации $A \to (B + C) = (A \to B) + (A \to C)$, получаем
   $Z_{124} \to (Z_1 + Z_2 + A) = (Z_{124} \to Z_1) + (Z_{124} \to Z_2) + (Z_{124} \to A)$
7) представим числа в двоичной системе счисления:
   124 = 1111100_2, 1 =1_2, 2 =10_2
8) поскольку двоичная запись чисел 1 и 2 содержит единичные биты, которых нет в наборе единичных битов числа 124, имеем
   $Z_{124} \to Z_1 = 0,\ Z_{124} \to Z_2 = 0$
   в том смысле, что найдутся такие значения x, при которых эти выражения ложны.
9) тогда для истинности заданного выражения остаётся обеспечить истинность $Z_{124} \to A$ при всех x, а это условие будет выполняться тогда и только тогда, когда все единичные биты двоичной записи числа a входят во множество единичных битов числа 124 = 1111100_2; таким образом, минимальное подходящее положительное значение a – 2^2 = 4, а максимальное – 124.

Ответ: 4.

3 *Введём выражение M & K, обозначающее поразрядную конъюнкцию M и K (логическое «И» между соответствующими битами двоичной записи). Определите наименьшее натуральное число a, такое что выражение*

((x & 28 ¹ 0) ∨ (x & 45 ¹ 0)) → ((x & 48 = 0) → (x & a ¹ 0))

тождественно истинно (то есть принимает значение 1 при любом натуральном значении переменной x)?

Решение:

1) Введём обозначения:
   Z_{28} **=** (x & 28 = 0), Z_{45} **=** (x & 45 = 0), Z_{48} **=** (x & 48 = 0), **A** = (x & a = 0)
2) перепишем исходное выражение и преобразуем его, используя свойство импликации:
   $$(\overline{Z}_{28} + \overline{Z}_{45}) \to (Z_{48} \to \overline{A}) = \overline{\overline{Z}_{28} + \overline{Z}_{45}} + (Z_{48} \to \overline{A}) = Z_{28} \cdot Z_{45} + \overline{Z}_{48} + \overline{A}$$
3) перейдем к импликации, используя закон де Моргана:
   $$Z_{28} \cdot Z_{45} + \overline{Z}_{48} + \overline{A} = \overline{Z_{48} \cdot A} + Z_{28} \cdot Z_{45} = (Z_{48} \cdot A) \to Z_{28} \cdot Z_{45}$$
4) преобразуем выражение в правой части по формуле $Z_K \cdot Z_M = Z_{K\ \mathrm{or}\ M}$, выполнив поразрядную дизъюнкцию (операцию ИЛИ):
   28 = 011100
   45 = 101101

28 or 45 = 111101 = 61

получаем $(Z_{48} \cdot A) \to Z_{61}$

5) для того, чтобы выражение $(Z_{48} \cdot A) \to Z_{61}$ было истинно для всех *x*, нужно, чтобы двоичная запись числа 48 **or** *a* содержала все единичные биты числа 61. Таким образом, с помощью числа a нужно добавить те единичные биты числа 61, которых «не хватает» в числе 48:

   48 = 110000
   a = **11*1
   61 = 111101

   биты, обозначенные звездочками, могут быть любыми.
6) поскольку нас интересует минимальное значение *a*, все биты, обозначенные звездочкой, можно принять равными нулю.
7) получается A = $2^3 + 2^2 + 2^0 = 13$

   Ответ: 13.

4.Дана система логических уравнений

$$F(x_1,x_2,x_3) \to F(x_4,x_5,x_6) = 1$$
$$F(x_4,x_5,x_6) \to F(x_7,x_8,x_9) = 1$$

где $x_1,x_2,\ldots,x_9$ – логические переменные, F – логическая функция трёх переменных, принимающая на $2^3 = 8$ всех возможных комбинациях входных переменных n_0 нулевых значений и n_1 единичных значений (очевидно что выполняется условие $n_0 + n_1 = 8$). При каком минимальном значении n_0 система имеет 320 решений?

Решение:

1) используем замену переменных

   $z_1 = F(x_1,x_2,x_3)$, $z_2 = F(x_4,x_5,x_6)$, $z_3 = F(x_7,x_8,x_9)$
2) свернём два уравнения в одно, перемножив левые и правые части:

   $(z_1 \to z_2)\cdot(z_2 \to z_3) = 1$
3) это элементарное уравнение, у которого существует 4 решения, имеющие структуру «все нули, потом все единицы» (см. http://kpolyakov.spb.ru/download/inf-2014-12a.pdf): битовая цепочка $Z = z_1z_2z_3$, составленная из логических переменных z_1, z_2, и z_3, может принимать значения 000, 001, 011 и 111
4) для перехода к исходным переменным вспомним, что в таблице истинности функции F(a,b,c), которую представляют переменные z_1, z_2, и z_3, есть n_0 нулевых значений и $n_1 = (8 - n_0)$ единичных значений
5) таким образом, каждый ноль в *Z*-решении увеличивает количество решений в n_0 раз, а каждая единица – в $n_1 = (8 - n_0)$ раз
6) например, количество X-решений, соответствующих *Z*-решениям равно

   000 $\to n_0^3$, 001 $\to n_0^2 n_1 = n_0^2(8-n_0)$,

   011 $\to n_0 n_1^2 = n_0(8-n_0)^2$, 111 $\to n_1^3 = (8-n_0)^3$
7) общее количество решений равно

   $N = n_0^3 + n_0^2(8-n_0) + n_0(8-n_0)^2 + (8-n_0)^3$,
8) раскрывая скобки и приводя подобные члены, получаем

   $N = 16n_0^2 - 128n_0 + 512$,
9) решая уравнение $N = 16n_0^2 - 128n_0 + 512 = 320$, получаем *n* = {2, 6}.

10) Ответ: 2

11) В силу симметрии множества *Z*-решений относительно 0 и 1 (множество не меняется при замене 0 на 1 и наоборот) имеем $N(n_0) = N(8 - n_0)$

12) Заметим, функция $N(n_0) = 16n_0^2 - 128n_0 + 512$ имеет минимум при

$$n_0^* = \frac{128}{2 \cdot 16} = 4$$

и максимальные значения на отрезке [0; 8] – на его концах.

Элементы теории алгоритмов

1. Два игрока, Петя и Ваня играют в следующую игру. На столе в кучке лежат фишки. На лицевой стороне каждой фишки написано двузначное натуральное число, обе цифры которого находятся в диапазоне от 1 до 4. Никакие две фишки не повторяются. Игра состоит в том, что игроки

поочередно берут из кучки по одной фишке и выкладывают в цепочку на стол лицевой стороной вверх таким образом, что каждая новая фишка ставится правее предыдущей и ближайшие цифры соседних фишек совпадают. Верхняя часть всех выложенных фишек направлена в одну

сторону, то есть переворачивать фишки нельзя. Например, из фишки, на которой написано 23, нельзя сделать фишку, на которой написано 32. Первый ход делает Петя, выкладывая на стол любую фишку из кучки. Игра заканчивается, когда в кучке нет ни одной фишки, которую можно добавить в цепочку. Тот, кто добавил в цепочку последнюю фишку, выигрывает, а его противник проигрывает. Будем называть партией любую допустимую правилами последовательность ходов игроков, приводящую к завершению игры. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит указать, какую фишку он должен выставить в любой ситуации, которая ему может встретиться при различной игре противника.

Пример. Пусть на столе в кучке лежат фишки: 11, 12, 13, 21, 22, 23

Пусть первый ход Пети 12. Ваня может поставить 21, 22 или 23. Предположим, он ставит 21. Получим цепочку 12-21. Петя может поставить 11 или 13. Предположим, он ставит 11. Получим цепочку 12-21-11. Ваня может поставить только фишку со значением 13. Получим цепочку 12-21-11-13. Перед Петей в кучке остались только фишки 22 и 23, то есть нет фишек, которые он мог бы добавить в цепочку. Таким образом, партия закончена, Ваня выиграл.

Выполните следующие три задания при исходном наборе фишек {12, 14, 21, 22, 24, 41, 42, 44}.

Задание 1.

а) Приведите пример самой короткой партии, возможной при данном наборе фишек. Если таких партий несколько, достаточно привести одну.

б) Пусть Петя первым ходом пошел 42. У кого из игроков есть выигрышная стратегия в этой ситуации? Укажите первый ход, который должен сделать выигрывающий игрок, играющий по этой стратегии. Приведите пример одной из партий, возможных при реализации выигрывающим игроком этой стратегии.

Задание 2

Пусть Петя первым ходом пошел 44. У кого из игроков есть выигрышная стратегия, позволяющая в этой ситуации выиграть своим четвертым ходом? Постройте в виде рисунка или таблицы дерево всех партий, возможных при реализации выигрывающим игроком этой стратегии. На рёбрах дерева указывайте ход, в узлах – цепочку фишек, получившуюся после этого хода.

Задание 3

Укажите хотя бы один способ убрать 2 фишки из исходного набора так, чтобы всегда выигрывал не тот игрок, который имеет выигрышную стратегию в задании 2. Приведите пример партии для набора из 6 оставшихся фишек.

Задание 1а.

1) партия заканчивается, когда цепочка закончилась на цифре X и не осталось ни одной фишки, которая бы начиналась с этой цифры;

2) меньше всего фишек заданного набора начинается с цифры 1 (только 12 и 14), поэтому самой короткой партией, вероятно, будет партия, которая заканчивается на цифре 1 (фишкой 21 или 41), при этом фишки 12 и 14 должны быть выставлены;

3) соединить эти фишки в цепочку можно с помощью фишек 21 или 41, таким образом, получается две возможных самых коротких партии:

12 – 21 – 14 – 41 и 14 – 41 – 12 – 21.

В ответе достаточно привести одну из них.

Задание 1б. (Идея решения – **А. Сидоров**).

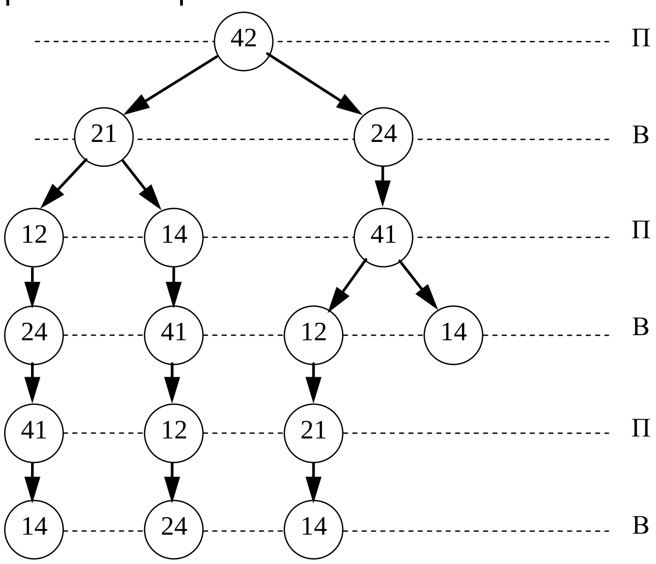
4) заметим, что эта игра напоминает «одностороннее домино», в котором фишки можно выставлять только одной стороной и наращивать цепочку можно тоже только с одной стороны;

5) среди фишек есть две особые – 22 и 44 («дубли») они служат для того, чтобы передать ход сопернику; если выставить дубль, оказавшись в проигрышной позиции, то эта проигрышная позиция «переходит» к сопернику

6) пока построим дерево без учёта дублей, то есть для набора фишек

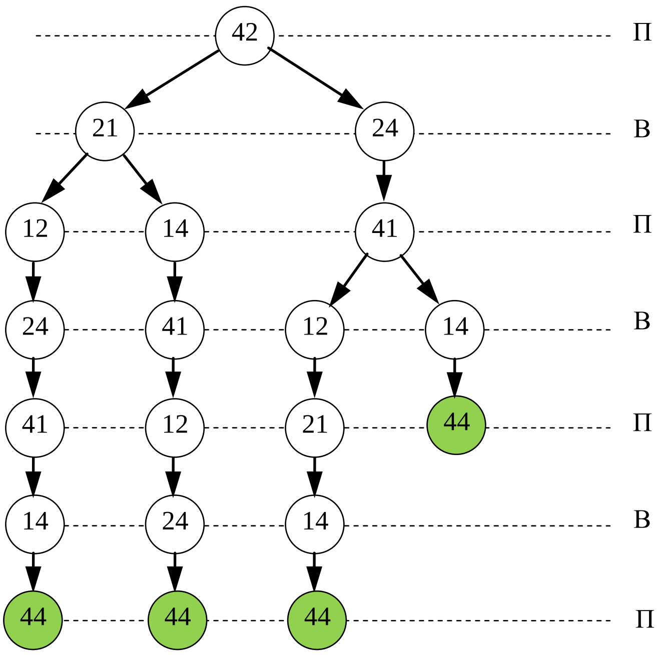
12, 14, 21, 24, 41 и 42

7) по условию Петя выставляет первым ходом фишку 42, дальнейшие варианты развития игры показаны на схеме:



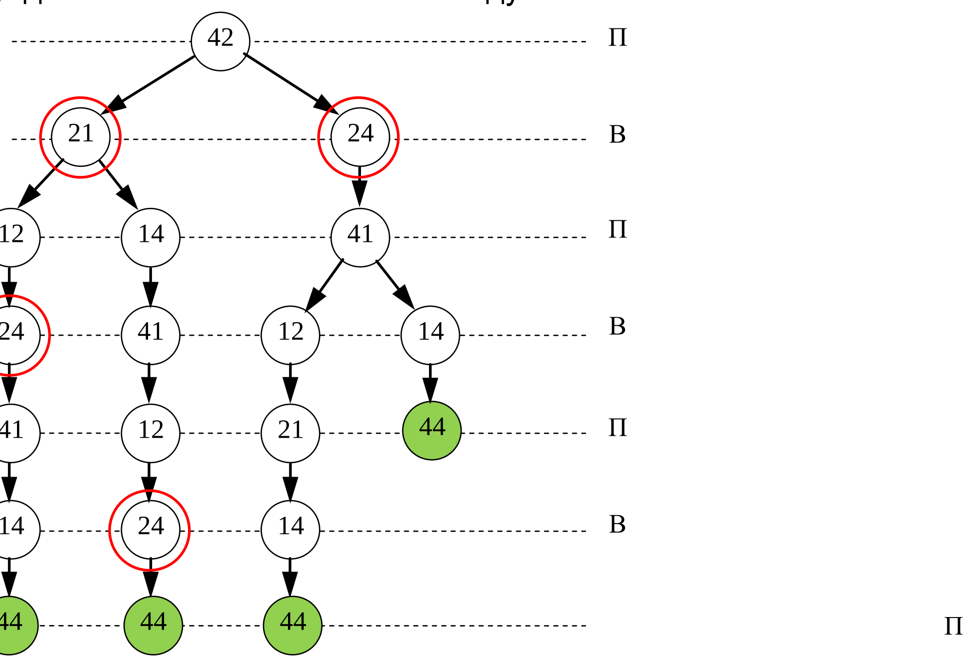
8) итак, мы видим, что если никто из игроков не выставляет дублей, то выигрывает Ваня во всех случаях, причем все партии заканчиваются на цифре 4

9) если Ваня в ходе игры не выставит дубль, то в конце каждой ветки Петя может выставить дубль 44 и выиграть:



10) поэтому теперь посмотрим, где Ваня может изменить игру дублями; Ване нет смысла ставить дубль 44, потому что во всех вариантах партий он уже есть (с выигрышем для Пети), так что выставление дубля 44 просто перемещает его в середину цепочки, не изменяя её длину

11) у Вани в распоряжении есть еще дубль 22; на следующем рисунке выделены ходы, где Ваня может поставить этот дубль:



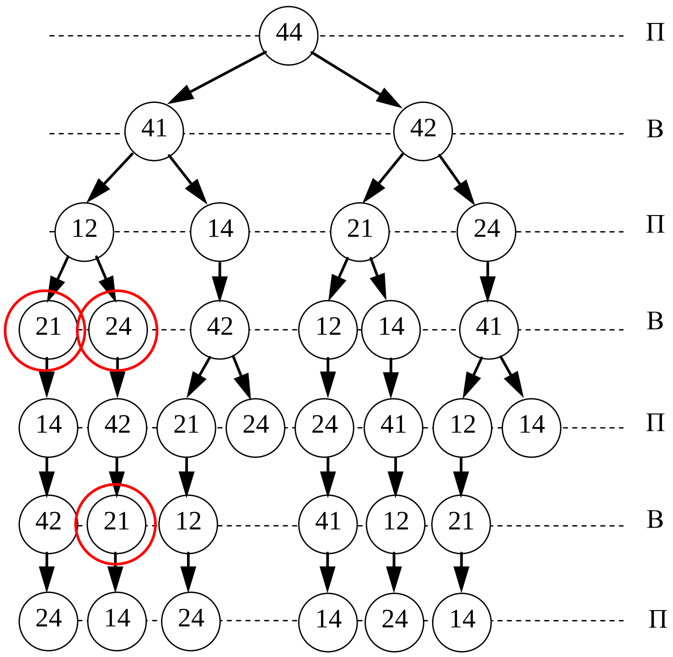
12) использование дубля 22 действительно изменяет игру, так как удлиняет цепочки на 1, при этом выигрывает Ваня:

а) Ваня может своим первым ходом выставить дубль 22, при этом он всегда выиграет

б) Ваня может первым ходом выставить фишку 21, при этом получив ход в позиции, когда текущая цепочка заканчивается на 2, он выставляет дубль 22 и выигрывает

Задание 2.

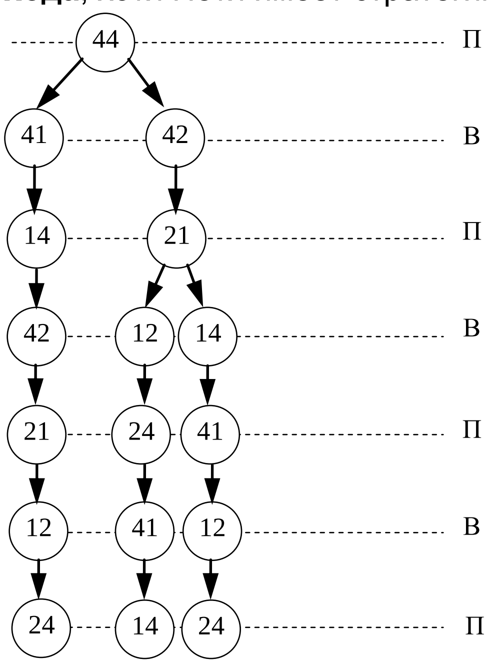
13) построим дерево игры для случая, когда Петя в самом начале ходит фишкой 44, «забыв» пока про дубль 22:



14) по дереву видим, что при игре без дубля 22 выигрывает Петя своим третьим или четвёртым ходом

15) Ваня может изменить ход игры дублем 22 только в выделенных узлах, то есть только тогда, когда Ваня сделает первый ход 41 и Петя ответит ходом 12; во всех остальных случаях выиграет Петя: если Ваня походит фишкой 41, Петя должен ответить ходом 14, а если Ваня походит фишкой 42, Петя может выбрать любое продолжение

16) при записи неполного дерева игры, доказывающего выигрыш Пети, нужно учесть, что по условию задачи нужно представить дерево **с выигрышем именно в 4 хода**, хотя Петя имеет стратегию выигрыша за 3 хода:

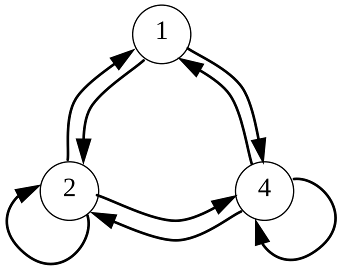


Задание 3.

17) в задании 2 выигрывает Петя, поэтому нужно убрать две фишки таким образом, чтобы всегда выигрывал Ваня

18) заметим, что нам нужно доказать, что Ваня выигрывает ВСЕГДА, при любом первом ходе Пети; это значит, что построение одного дерева (при конкретном первом ходе) ничего не доказывает

19) заметим, что последней цифрой цепочки для данного набора фишек всегда будет 1, 2 или 4, поэтому можно построить такой граф возможных переходов (например, ребро перехода 1 $\rightarrow$ 2 соответствует фишке 12, а петля у узла 2 – фишке 22):



по каждому ребру этого графа можно пройти только один раз (каждая фишка выставляется один раз)

20) по этому графу видно, что если убрать две петли, остается граф с одинаковой ситуацией для каждой из вершин: есть два контура, проходящие в разных направлениях через все узлы

21) итак, если убрать два дубля, то всегда будут выставлены 4 или все 6 фишек, поскольку 4 и 6 – чётные числа, то всегда выиграет Ваня, потому что он делает все ходы с чётными номерами; например, при первом ходе Пети 12 возможны партии:
12 – 21 – 14 – 41 или 12 – 24 – 41 – 14 – 42 – 21.

Языки программирования

1. На вход программе подаются сведения о номерах школ учащихся, участвовавших в олимпиаде. В первой строке сообщается количество учащихся N, каждая из следующих N строк имеет формат:

<Фамилия> <Инициалы> <номер школы>

где <Фамилия> – строка, состоящая не более чем из 20 символов, <Инициалы> – строка, состоящая из 4-х символов (буква, точка, буква, точка), <номер школы> – не более чем двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом. Пример входной строки:

Иванов П.С. 57

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран информацию, из какой школы было меньше всего участников (таких школ может быть несколько). При этом необходимо вывести информацию только по школам, пославшим хотя бы одного участника. Следует учитывать, что N>=1000.

Как правильно понимать условие?

6) на первый вопрос – как именно вводятся данные – находим ответ в самом начале условия: вроде бы «дежурная» фраза «на вход программе подаются…» означает, что данные нужно читать не из файла, а со стандартного входного потока; это, в свою очередь, значит, что можно использовать привычные операторы **read** (**readln**), предполагая, что кто-то вводит эти данные с клавиатуры вручную

7) итак, сначала вводится количество записей в файле **N**, а затем **N** строк с информацией; заметим, что из всей этой информации нас интересует (в каждой строке) только номер школы, остальное можно просто отбрасывать
8) номер школы стоит после второго пробела в строке
9) «*<номер школы> – не более чем двузначный номер*» – крайне важная информация; собственно, только она и позволяет найти хорошее решение задачи; это значит, что школ не более 99!
10) что означает выражение «как можно более эффективная программа»?
    - прежде всего, данные читаются только один раз, за один проход, нельзя «вернуться» и прочитать что-то вновь
    - в программе не выполняются никакие лишние действия
    - используемые алгоритмы имеют минимальную сложность (см. выше)
    - расходуется минимальный возможный объем памяти; например, чтобы найти количество отрицательных элементов массива, не нужно вводить второй массив; если нам достаточно держать в памяти одну введенную строку, не нужно одновременно хранить все прочитанные строки
11) зачем нужно уточнение «*N>=1000*»? этим авторы задачи намекают на то, что не нужно считывать все данные в оперативную память, а потом уже их обрабатывать; основная обработка должна быть сделана сразу, в том же цикле, где читаются входные данные
12) мы будем считать, что в исходных данных нет ошибок (так принято на олимпиадах и экзаменах), иначе обработка разнообразных ошибок будет составлять основную часть программы

Решение:

1) по условию, единственная информация, которая нам нужна в итоге для вывода результата – это количество участников по каждой школе
2) так как номер школы состоит (по условию!) не более, чем из двух цифр, всего может быть не более 99 школ (с номерами от 1 до 99)
3) поэтому можно ввести массив **C** из 99 элементов; для всех **k** от 1 до 99 элемент **C[k]** будет ячейкой-счетчиком, в которой накапливается число участников от школы с номером **k**; сначала во все элементы этого массива записываются нуль (обнуление счетчиков):
   ```
   for k:=1 to 99 do C[k]:=0;
   ```
   во многих системах программирования на Паскале все глобальные переменные автоматически обнуляются, и таким образом, этот цикл ничего не дает; однако на всякий случай нужно продемонстрировать эксперту, который будет проверять часть **C** вашей работы, что вы понимаете суть дела («счетчик необходимо сначала обнулить»)
4) основной цикл обработки вводимых строк можно записать на псевдокоде так:
   ```
   for i:=1 to N do begin
    { читаем очередную строку }
    { определяем номер школы k }
    C[k] := C[k] + 1; { увеличиваем счетчик k-ой школы }
   end;
   ```
5) поскольку данные вводятся в виде символьной строки, нужно выделить в памяти переменную **s** типа **string**

6) для чтения очередной строки будем использовать оператор **readln**
7) остается понять, как выделить из строки номер школы; по условию он закодирован в последней части строки, после второго пробела; значит, нужно найти этот второй пробел, вырезать из строки весь «хвост» после этого пробела, и преобразовать его из символьного формата в числовой
8) чтобы найти первый пробел и «отрезать» первую часть строки с этим пробелом, можно использовать команды

```
p := Pos(' ', s);
s := Copy(s, p+1, Length(s)-p);
```

первая команда определяет номер первого пробела и записывает его в целую переменную **p**, в вторая – записывает в строку **s** весь «хвост», стоящий за этим пробелом, начиная с символа с номером **p+1**; длина хвоста равна **Length(s)-p**, где **Length(s)** – длина строки;
9) поскольку нас интересует часть после **второго** пробела, эти две строчки нужно повторить два раза, в результате в переменной **s** окажется символьная запись номера школы;
10) заметим, что можно избежать дублирования двух строк, «свернув» их во внутренний цикл, но это вряд ли сильно упростит запись:

```
for k:=1 to 2 do begin
 p := Pos(' ', s);
 s := Copy(s, p+1, Length(s)-p);
end;
```

11) в пп. 8-10 описан достаточно общий метод, при котором инициалы могут быть любой длины, (но без пробела); в данном случае в условии четко сказано, что инициалы представляют собой именно 4 символа (буква, точка, буква, точка), поэтому можно найти первый пробел, а затем взять «хвост», который идет через 6 символов от него:

```
p := Pos(' ', s);                  или   p := Pos(' ', s);
s := Copy(s,p+6,Length(s));        так   Delete(s, 1, p+5);
```

12) для преобразования номера школы из символьного вида в числовой можно использовать функцию **Val**:

```
Val(s, k, r);
```

эта процедура (*Turbo Pascal, Borland Pascal, PascalABC,* среда *АЛГО*) преобразует символьную строку **s** в числовое значение **k**; с помощью переменной **r** обнаруживается ошибка: если раскодировать число не удалось (в строке не число), в **r** будет записан нуль (здесь мы не будем обрабатывать эту ошибку, полагая, что все данные правильные);
если вы работаете на *ПаскалеABC* (никто не может вам запретить написать, что этот так), вместо **Val** можно использовать более удобную и понятную функцию **StrToInt**:

```
k := StrToInt(s);
```

13) таким образом, основной цикл выглядит так:

```
for i:=1 to N do begin
 readln(s); { читаем очередную строку }
  { выделяем часть после второго пробела }
 p := Pos(' ', s);
```

```
 Delete(s, 1, p+5);
  { определяем номер школы k }
 Val(s, k, r);
 C[k] := C[k] + 1; { увеличиваем счетчик k-ой школы }
end;
```

14) дальше стандартным алгоритмом определяем в массиве **C** минимальный элемент **Min**, не учитывая нули (школы, из которых не было участников):

```
Min := N;
for k:=1 to 99 do
 if (C[k] <> 0) and (C[k]<Min) then Min := C[k];
```

здесь интересна первая строчка, **Min:=N**: по условию всего было **N** участников, поэтому минимальное значение не может быть больше **N**; обратите внимание, что привычный вариант (который начинается с **Min:=C[1]**) работает неверно, если из первой школы не было ни одного участника

15) и выводим на экран номера всех школ (обратите внимание – **номера!**), для которых **C[k]=Min**:

```
for k:=1 to 99 do
 if C[k] = Min then writeln(k);
```

16) остается «собрать» программу, чтобы получилось полное решение; максимальное количество школ мы задали в виде константы **LIM**:

```
const LIM = 99;
var C:array[1..LIM] of integer;
  i, p, N, k, r, Min: integer;
  s:string;
begin
for k:=1 to 99 do C[k]:=0;
readln(N);
for i:=1 to N do begin
 readln(s); { читаем очередную строку }
  { выделяем часть после второго пробела }
 p := Pos(' ', s);
 Delete(s, 1, p+5);
  { определяем номер школы k }
 Val(s, k, r);
 C[k] := C[k] + 1; { увеличиваем счетчик k-ой школы }
end;
Min := N;
for k:=1 to LIM do
 if (C[k] <> 0) and (C[k]<Min) then Min := C[k];
for k:=1 to LIM do
 if C[k] = Min then writeln(k);
end.
```

Темы индивидуальных заданий

по дисциплине «Практикум по решению задач повышенной сложности по информатике»

Разработать или подобрать задачи повышенной сложности на заданную тему. Задач должно быть не менее трёх. Решить все задачи. Оформить отчёт в электронном виде, содержащий условия и решения всех задач.

Темы заданий

1. Системы счисления
2. Кодирование информации.
3. Двоичное кодирование.
4. Моделирование.
5. Построение алгоритмов
6. Быстрая сортировка.
7. Выигрышные стратегии
8. Логические задачи
9. Графы. Поиск в глубину.
10. Графы. Поиск в ширину.