Методические указания по Отраслевой олимпиаде школьников «Газпром», профиль «информатика»

Учебное пособие для подготовки к олимпиаде

Под редакцией Ю.Б.Буркатовской, В.В.Видмана

Тематика заданий

1. Комбинаторика.
2. Обработка данных, программирование.
3. Логика.
4. Логическая цепочка.
5. Графы.
6. Шифрование.
7. Системы счисления.
8. Булева алгебра.
9. Кодирование и декодирование.
10. Количество информации.

Задание 1. Комбинаторика

Задача 1. Для проверки нефтяной отрасли была создана комиссия, которая должна была в максимально быстрые сроки проверить ряд предприятий. Всего необходимо проверить 5 нефтедобывающих предприятий и 7 нефтеперерабатывающих завода. За одну неделю комиссия может проверить максимум 1 добывающее и 2 перерабатывающих предприятия. Каждую неделю комиссия решает, на какие предприятия поехать.

Сколько вариантов выбора объектов есть у комиссии в начале второй недели?

Решение задачи 1.

Всего необходимо проверить 5 нефтедобывающих предприятий и 7 нефтеперерабатывающих завода. После первой недели их осталось 4 добывающих (так как 5−1=4, т.е. на первой неделе проверили 1 добывающее предприятие, и мы его исключаем из общего числа) и 5 перерабатывающих (7−2=5). Для решения можно воспользоваться формулой числа сочетаний:

$$C_4^1 = 4$$

$$C_5^2 = \frac{5 * 4 * 3!}{(5-2) * 2!} = \frac{5 * 4}{2} = 10$$

Иначе, из 4 добывающих мы можем четырьмя способами взять по одному (1,2,3,4). Из 5 перерабатывающих мы можем 10 способами взять по два (1 и 2, 1 и 3, 1 и 4, 1 и 5, 2 и 3, 2 и 4, 2 и 5, 3 и 4, 3 и 5, 4 и 5)

Каждое из 4 добывающих предприятий может комбинироваться с каждой из 10 пар перерабатывающих. Перемножим эти числа и получим количество вариантов выбрать объекты для проверки на вторую неделю. 4 * 10 = 40.

Ответ. 40 вариантов.

Задачи в других вариантах аналогичны, поэтому приводится решение только для одного варианта.

Задание 2. Обработка данных и программирование

Задача 2. Давление газа, подаваемого из скважины, измеряется каждые 15 минут, в результате получается последовательность значений – вещественных чисел в диапазоне от 0 до 20. Спадом давления будем называть элемент в этой последовательности, который меньше как предыдущего, так и последующего значения. При нормальной работе скважины спад не должен быть меньше среднего значения давления на всем промежутке наблюдений, но без учета спадов, более, чем в k раз. Если такое происходит, это считается опасной ситуацией. Требуется написать программу, подсчитывающую количество опасных ситуаций на всем участке измерений. Известно, что измерения производятся не более суток. Гарантируется, что имеется хотя бы один спад.

Входные данные. В первой строке содержится целое число n - количество измерений. Во второй строке содержится целое число k. В следующих строках содержится n вещественных чисел – результаты измерений.

Выходные данные. Целое число – количество опасных ситуаций на всем участке измерений.

Пример входных данных.

15
2
8.24
9.12
3.93
8.99
6.40
9.47
3.51
4.44
9.35
4.75
4.15
9.46
6.82
5.65
3.64

Пример выходных данных (для данного примера входных данных)

1

Пояснение. В последовательности 4 спада (3.93, 6.40, 3.51, 4.15, выделены красным)

8.24
9.12
3.93

8.99
6.40
9.47
3.51
4.44
9.35
4.75
4.15
9.46
6.82
5.65
3.64

После исключения спадов остается 11 значений, среднее арифметическое которых равно 7.26 (с точностью до двух знаков после запятой). По условиям, критическая ситуация возникает, если значение спада меньше, чем 7.22, деленное на два, т.е., меньше 3.63. В последовательности один такой спад – 3.51.

Решение задачи 2.

Данные о давлении могут быть сохранены в массиве вещественных чисел. Поскольку измерения проводятся не более суток каждые 15 минут, максимальный размер массива будет 24×4=96. Рассматривается заданное число наблюдений n≤96. Чтобы определить, является ли элемент массива с индексом i спадом, нужно проверить, меньше ли он обоих соседних элементов, то есть элементов с индексами i−1 и i+1. Таким образом, первое и последнее значение не могут быть спадами, а остальные элементы не являются спадами, если не меньше хотя бы одного соседнего элемента. Необходимо вычислить среднее значение элементов, не являющихся спадами, для этого требуется знать их сумму и количество. Таким образом, за первый проход по массиву для каждого элемента выясняем, является ли он спадом. Если нет, добавляем его к накапливаемой сумме и увеличиваем счетчик на единицу. После выполнения данного цикла вычисляем среднее значение элементов, не являющихся спадами, разделив их сумму на количество. Затем повторно проходим по массиву, опять проверяя для каждого элемента, является ли он спадом. Если да, то сравниваем его с вычисленным средним арифметическим значением, если оно превышает спад более, чем в k раз, увеличиваем счетчик таких элементов на единицу. Помним, что перед циклом все подсчитываемые счетчики и суммы обнуляются. В конце работы программы выводим значение счетчика.

Ниже приведена программа на языке Pascal.

```
Program Problem_2;
var  i,n,k,count: integer;
     sum,sa: real;
     A: array[1..96] of real;
begin
     readln(n);
     readln(k);
```

```
    for i:=1 to n do
        readln (A[i]);
    sum:=0;
    count:=0;
    for i:=2 to (n-1) do
        if ((A[i]>=A[i-1])or(A[i]>=A[i+1])) then
        begin
            sum:=sum+A[i];
            count:=count+1;
        end;
    sum:=sum+A[1]+A[n];
    count:=count+2;
    sa:=sum/count;
    count:=0;
    for i:=2 to (n-1) do
        if ((A[i]<A[i-1])and(A[i]<A[i+1])) then
            if(k*A[i]<sa) then
                count:=count+1;
    writeln(count);
end.
```

Ниже приведена программа на языке C++.

```
#include<iostream>
using namespace std;
int main()
{
    int i, n, k, count;
    float sum, sa, A[96];
    cin >> n >> k;
    for(i=0;i<n;i++)
        cin >> A[i];
    sum=0;
    count=0;
    for(i=1;i<n-1;i++)
        if ((A[i]>=A[i-1])||(A[i]>=A[i+1]))
        {
            sum+=A[i];
            count++;
        }
    sum+=(A[0]+A[n-1]);
    count+=2;
    sa=sum/count;
    count=0;
    for(i=1;i<n-1;i++)
```

```
            if ((A[i]<A[i-1])&&(A[i]<A[i+1]))
                  if(k*A[i]<sa)
                        count++;
      cout << count;
      return 0;
}
```

Ниже приведена программа на языке Python.

```
n=int(input())
k=int(input())
A=[]
for i in range(n):
     A.append(float(input()))
sum=0;
count=0;
for i in range(1,n-1):
      if ((A[i]>=A[i-1])or(A[i]>=A[i+1])):
            sum+=A[i]
            count+=1
sum+=(A[0]+A[n-1]);
count+=2;
sa=sum/count;
count=0;
for i in range(1,n-1):
      if ((A[i]<A[i-1])and(A[i]<A[i+1])):
            if(k*A[i]<sa):
                  count+=1
print(count)
```

Заметим, что приведенная программа не является единственным вариантом, возможны и другие решения.

В других вариантах задача 2 имеет другие условия, но по сложности и по идее решения все задачи близки, поэтому мы приводим решение только для одного варианта.

Задание 3. Логика

Задача 3.1. Геологи Джон и Тирион вместе брали пробы с месторождения, но не могут вспомнить точную дату. Общими усилиями они вычислили список возможных дат: 15, 16 или 19 мая; 17 или 18 июня; 14 или 16 июля; 14, 15 или 17 августа. Их коллега геолог Серсея любит пошутить, а также точно знает дату. Она сообщила Джону месяц, и только месяц, а Тириону число, и только число. «Теперь ты можешь догадаться?» — спросила она у Джона.

Джон: Я не знаю, но я знаю, что Тирион тоже не знает.

Тирион: Сначала я не знал, но теперь знаю.

Джон: Что ж, теперь и я знаю!

В какой день геологи брали пробы с месторождения?

Решение задачи 3.1.

Для наглядности составим таблицу, в которой цветом выделим возможные даты.

	май	июнь	июль	август
14				
15				
16				
17				
18				
19				

Видно, что в мае и июне есть даты, которых нет в других месяцах, это 19 (май) и 18 (июнь). Поэтому, если бы Тириону сказали одно из этих чисел, он бы сразу дал правильный ответ. Поэтому, если Джон уверен, что Тирион не знает верного ответа, значит, пробы брались в июле или августе.

Из ответа Джона Тирион понимает, что месяц – июль или август, и говорит, что знает ответ. Значит, число не 14, поскольку в списке дат есть 14 июля и 14 августа. Если у Тириона число 15 или 17, то месяц – август, если 16 – июль.

Поскольку после ответа Тириона Джон говорит, что тоже знает ответ, значит, месяц – не август, поскольку в августе есть два возможных дня. Остается единственный вариант – 16 июля.

Задача также известна как «День рождения Шерил», посмотреть ее можно, например, здесь https://lifehacker.ru/logicheskie-zadachi/ .

Ответ. 16 июля.

Задача 3.2. Четыре студента геологоразведочного факультета (Арья, Джон, Серсея и Иннокентий) выполняют лабораторные работы. Задания раздаются по круговой системе (каждый студент выполняет одно задание в паре с каждым другим студентом). Если оба студента работают одинаково хорошо, оба получают по одному баллу. Если один выполнил основную часть задания, он получает два балла, а его партнер - ни одного. После каждой совместной работы преподаватель формирует рейтинг учащихся. Известно, что Серсея не получила ни одного балла по итогам последнего дня работы в паре Серсея-Иннокентий. Тем не менее, в итоге у Серсеи больше всех баллов, а Иннокентий не улучшил свое положение в рейтинге. Какие баллы получила за совместную работу пара Джон-Иннокентий?

Решение задачи 3.2. Прежде всего, заметим, что из четырех студентов можно организовать шесть пар. В каждой паре по результатам работы разыгрываются два балла, т.е. общее число баллов равно 12. Таким образом, если Серсея набрала баллов больше всех, при том, что во время совместной работы с Иннокентием она не набрала ни одного балла, то единственный вариант – в паре с Арьей и Джоном она каждый раз набирала по два балла, и в сумме имеет 4 балла. Остальные студенты в сумме имеют 8 баллов, причем балл каждого студента не больше трех. Единственный вариант – баллы студентов равны 2, 3 и 3. Если бы балл Иннокентия был 3, то до совместной работы с Серсеей он был бы 1, т.е., до этого он был последним в рейтинге, а после работы с Серсеей делил бы 2−3 место, т.е. улучшил бы свое положение в рейтинге. Значит, Иннокентий имел 0 баллов, после работы с Серсеей имеет 2 балла, и остается последним в рейтинге. Таким образом, в паре Джон-Иннокентий Джон получил два балла, а Иннокентий – ни одного.

Итоги работы можно представить в виде таблицы, где строки и столбцы соответствуют студентам, в строке указаны баллы студента при совместной работе со студентами, чьи имена указаны в столбцах.

	Арья	Джон	Иннокентий	Серсея
Арья		1	2	0
Джон	1		2	0
Иннокентий	0	0		2
Серсея	2	2	0	

Ответ. Джон – 2 балла, Иннокентий – 0 баллов.

Задача 3.3. На семинаре по новым методам разработки скважин встретились сотрудники различных газовых компаний. Поскольку данный семинар проводится не впервые, каждый человек знаком не менее, чем с половиной присутствующих. В плане семинара стоит «круглый стол», для которого нужно выбрать четырех человек. Можно ли выбрать четырех человек и рассадить за на самом деле круглым столом так, что каждый будет сидеть со своими знакомыми? Обоснуйте свой ответ.

Решение задачи 3.3. Предположим, что на семинаре есть два человека, незнакомых друг с другом (если это не так, то все участники попарно знакомы, и можно выбрать любых четырех человек и рассадить их в произвольном порядке). Назовем их A и B. Пусть $N(A)$ – число знакомых A, $N(B)$ – число знакомых B. Посадим их друг напротив друга. Если всего на семинаре n человек, то $N(A) \geq n/2$, $N(B) \geq n/2$, причем, поскольку A и B незнакомы, все их знакомые принадлежат множеству из $n-2$ человек, которые еще не сидят за столом. Пусть $N(A \cup B)$ – количество человек, знакомых с A или B, $N(A \cap B)$ – количество человек, знакомых с A и B одновременно, тогда

$N(A \cup B) = N(A) + N(B) - N(A \cap B)$;

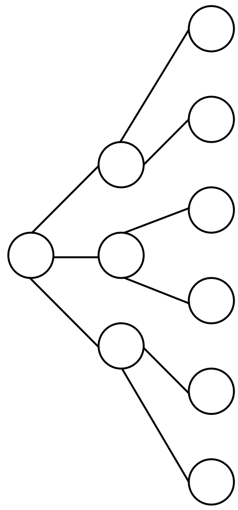
$N(A \cup B) \leq n - 2$;

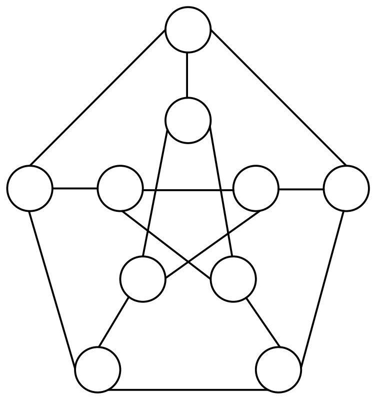
$N(A) + N(B) \geq n$.

Отсюда следует, что $N(A \cap B) \geq 2$, т.е. есть как минимум два человека, знакомых A и B одновременно, их можно посадить за круглый стол на оставшиеся места.

Задача 3.4. Газовые скважины на месторождении связаны дорогами, причем к каждой скважине походит не более, чем три дороги. Сотрудники компании могут добраться от любой до любой скважины, посетив по пути не более одной скважины. Каково максимальное число скважин на месторождении? Обоснуйте свой ответ.

Решение задачи 3.4. Заметим, что скважин не может быть больше 10, т.к. от первой скважины существуют дороги до не более чем трех скважин, а от каждой из этих скважин есть дороги к не более чем двум новым скважинам (это иллюстрирует левый рисунок). Осталось показать, что возможно связать 10 скважин дорогами при заданных условиях. Эту возможность иллюстрирует диаграмма справа (граф Петерсена).





Ответ. 10 скважин.

Задачи из остальных вариантов также можно решить с помощью логических рассуждений. Также для многих из них подходят методы теории графов. Часть задач можно найти, например, в книге О.И.Мельникова «Занимательные задачи по теории графов».

Задание 4. Логическая цепочка

Задача 4. Находясь на своей обычной смене, Брон занимался своими рутинными обязанностями (он должен следить за датчиками давления в трубопроводе), как вдруг мониторы начали моргать, и потом появилась надпись на экране, читая которую, он понял, что на компьютере появился вирус. Поверх всех окон висело сообщение о том, что нужно отправить деньги на такой-то номер, а взамен получить код для разблокировки. По инструкции он должен немедленно сообщить об этом, но тогда его точно уволят, потому как сочтут виновным в появлении вируса на компьютере. Он решил попытаться самостоятельно найти ответ.

Представлен ряд чисел. Необходимо расшифровать числа и дописать недостающее:

198 240 306 402 530 1243 3012 21100 …

Решение задачи 4. В данном наборе все числа представляют одно и тоже число, но в разных системах счисления. Слева направо числа написаны, начиная с числа в десятичной системе счисления и заканчивая троичной. Число, которое нужно дописать, это число 198, переведенное в двоичную систему счисления.

198/2 = 99 остаток = 0

99/2 = 49 остаток = 1

49/2 = 24 остаток =1

24/2 = 12 остаток = 0

12/2 = 6 остаток = 0

6/2 = 3 остаток = 0

3/2 = 1 остаток = 1

1/2 = 0 остаток = 1

Ответ – это число, составленное из остатков от деления записанное с конца 11000110. Также можно было решить эту задачу, переведя в двоичную систему число из восьмеричной или четверичной системы. Для этого достаточно заменить цифру ее двоичным кодом длины 3 для восьмеричной, и 2 – для четверичной системы.

Ответ. 11000110.

Задание 5. Графы

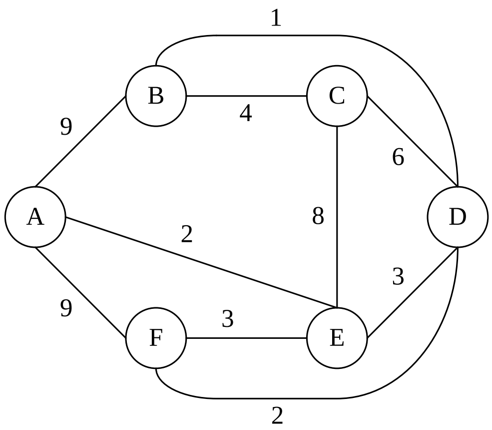
Задача 5. На месторождении шесть пунктов, в которых расположены скважины для добычи газа. Для их обслуживания запланировано две ремонтных бригады, базы которых должны быть расположены в различных пунктах. Если возникает необходимость в ремонте, вызывается бригада с ближайшей базы, при этом возможен проезд через другой пункт. Расположите базы так, чтобы максимальное расстояние, которое проезжает бригада до скважины, было минимально. В таблице указаны длины дорог между пунктами, пустая клетка означает, что дороги не существует (кроме клеток на главной диагонали). В ответе нужно указать, в каких пунктах располагаются базы, и какие скважины обслуживает каждая база, а также максимальную длину пути ремонтной бригады.

	A	B	C	D	E	F
A		9			2	9
B	9		4	1		
C		4		6	8	
D		1	6		3	2
E	2		8	3		3
F	9			2	3	

Пример. Если расположить базы в пунктах A и B, то база A обслуживает скважины A (расстояние от A до A равно нулю), E (расстояние от A до E равно 2; база B обслуживает скважины B (расстояние равно 0), C (расстояние 4), D (расстояние 1) , F (расстояние равно 3, через скважину D). Максимальное расстояние равно 4.

Решение задачи 5

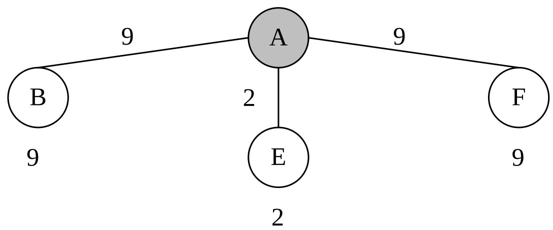
Приведенную информацию о скважинах и дорогах можно представить в виде диаграммы графа. Вершинам соответствуют скважины, ребрам – дороги, числа возле ребер обозначают длины дорог. Если вы не знакомы с математическим понятием «граф», можно начать с Википедии, либо любого другого источника.



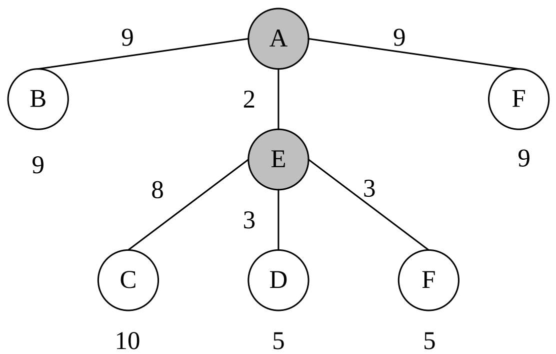
Найдем кратчайшие расстояния между скважинами. Не всегда выгодно передвигаться напрямую по ребру, иногда путь через промежуточную скважину оказывается короче. Например, от C до D выгоднее добираться по пути CBD, итоговая длина составит 4+1=5, тогда как длина дороги CD равна 6. Кратчайшие расстояния можно найти визуально по диаграмме графа, либо используя специальные алгоритмы.

Для примера рассмотрим один из классических алгоритмов поиска кратчайшего пути в графе – алгоритм Дейкстры, который может быть использован для графов с положительными весами ребер. Обучающее видео по алгоритму можно посмотреть, например, по ссылке https://foxford.ru/wiki/informatika/algoritm-deykstry. Также можно рекомендовать статью по ссылке https://habr.com/ru/post/111361/, с подробным разбором примера.

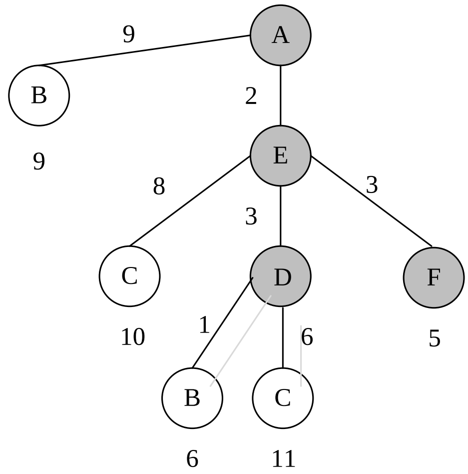
Выбираем стартовую вершину A и рассматриваем все смежные с ней (т.е. соединенные ребром) вершины. Стартовую вершину окрашиваем.



На данный момент найдены пути до вершин B, E, F, их длины 9, 2 и 9 соответственно (длины путей указаны под вершинами). Длина пути до вершины E минимальна, значит, найден кратчайший путь (если добираться до E через вершины B или F, то длина пути будет не меньше 9. Окрашиваем вершину E и рассматриваем все неокрашенные смежные с ней вершины.

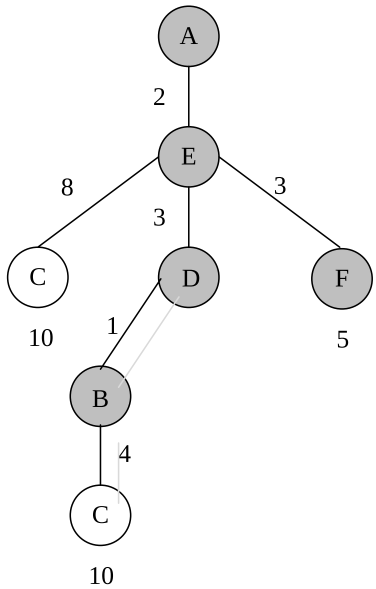


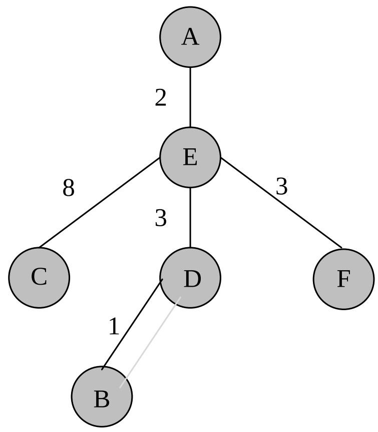
Из неокрашенных вершин минимальную длину пути имеют D и F, окрашиваем их и рассматриваем смежные вершины. Обращаем внимание, что прежний путь до вершины F (длины 9) исключается из рассмотрения.



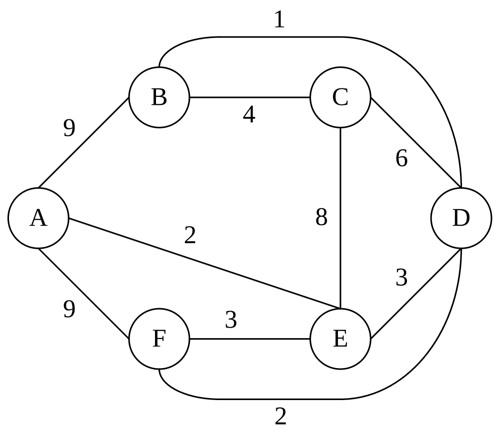
Исключаем из рассмотрения пути AB длины 9 и AEDC длины 11, поскольку имеются более короткие пути AEDB длины 6 и AEC длины 10. Минимальную длину пути из неокрашенных вершин имеет вершина B. Окрашиваем ее и рассматриваем смежные с ней неокрашенные вершины.

Неокрашенной осталась лишь вершина C, до нее на данный момент имеется два пути длины 10. Можно выбрать любой. Работа алгоритма закончена. Построено дерево кратчайших путей.





Далее мы не расписываем процесс поиска кратчайших путей, а лишь приводим итоговую таблицу с их длинами. Правильность вычислений можно проверить самостоятельно по диаграмме.



	A	B	C	D	E	F
A	0	6	10	5	2	5
B	6	0	4	1	4	3
C	10	4	0	5	8	7
D	5	1	5	0	3	2
E	2	4	8	3	0	3
F	5	3	7	2	3	0

Расстоянием вершина-вершина $d(x, y)$ в графе называется длина кратчайшего пути от вершины x до вершины y. В таблице приведены именно эти расстояния.

Максимальным удалением $r(x)$ от вершины x называется расстояние от вершины x до самой удаленной от нее вершины. Максимальное удаление легко найти из таблицы расстояний – это максимальное число в строке:

$$r(x) = \max_y d(x, y).$$

Центром графа называется та его вершина, для которой максимальное удаление минимально. В таблице ниже приведены максимальные удаления, центр – вершина D.

	A	B	C	D	E	F	r(x)
A	0	6	10	5	2	5	10
B	6	0	4	1	4	3	6
C	10	4	0	5	8	7	10
D	5	1	5	0	3	2	5
E	2	4	8	3	0	3	8
F	5	3	7	2	3	0	7

Таким образом, поиск центра не представляет трудностей, и если бы стояла задача найти базу для одной ремонтной бригады, она располагалась бы на скважине D. Но по условиям нам требуется две базы, и обслуживает скважину всегда ближайшая база. Для математического описания этой задачи вводится расстояние множество-вершина. Пусть X – множество вершин.

Расстоянием множество-вершина $d(X, y)$ называется минимальное из расстояний от вершины из множества X до вершины y

$$d(X, y) = \min_{x \in X} d(x, y).$$

Эти расстояния можно также найти из таблицы, если задано множество X, достаточно рассмотреть только строки, соответствующие вершинам этого множества, и выбрать минимум в каждом столбце. Ниже приведем пример для $X = \{A, B\}$.

	A	B	C	D	E	F	r(x)
A	0	6	10	5	2	5	10
B	6	0	4	1	4	3	6
X={A,B}	0	0	4	1	4	3	4

Таким образом, ставится задача поиска множества X с минимальным показателем r(X). В теории графов эта задача известна, как задача поиска *кратного центра*. Она относится к классу задач неполиномиальной сложности (NP), т.е., требует перебора всех возможных решений, либо может быть решена с помощью эвристических методов, не гарантирующих точного решения. В нашем примере решением может быть множество {A,B} или {B,E}, максимальное удаление от этих множеств равно 4, и не существует множества с максимальным удалением 3.

Ответ. Ремонтные бригады располагаются в пунктах $\{A, B\}$. Бригада A обслуживает скважины A, E, бригада B – пункты B, C, D, F. Максимальное расстояние равно 4.

Задачи в других вариантах аналогичны, поэтому приводится решение только для одного варианта.

Задание 6. Шифрование

Задача 6. Информация, поступающая с месторождения, рассматривается как набор байтов. Для защиты от промышленного шпионажа каждый байт шифруется с использованием следующих операций:

1) Циклический сдвиг влево на одну позицию (например, байт 10011100 преобразуется в байт 00111001);
2) Инверсия (например, инверсия байта 10011100 есть 01100011);
3) Побитовое сложение по модулю два с маской (например, если байт равен 10011100, а маска – 11010010, то результат их сложения 01001110).

Каждый байт задает целое число, которое преобразуется после кодирования в другое целое число. Известно, что для шифрования использовались описанные выше операции со следующими ограничениями:

1) Каждая операция использовалась не более одного раза;
2) В маске нет двух нулей, идущих подряд, но нули присутствуют.

Предложите алгоритм шифрования, который преобразует число 81 в число 55, и соответствующий алгоритм дешифрования. Возможно ли, что существует число, которое не меняется после предложенного шифрования?

Решение задачи 6

Будем использовать следующие обозначения:

$\bar{X}$	Инверсия двоичного числа X
$X \oplus Y$	Сумма по модулю два двоичных чисел X и Y
$X \ll 1$	Циклический сдвиг влево двоичного числа X на одну позицию
$X \gg 1$	Циклический сдвиг вправо двоичного числа X на одну позицию

Заметим, что, если сдвиг и инверсия выполняются подряд, то неважно, в каком порядке они выполняются. Для побитового сложения по модулю два и сдвига порядок, безусловно, важен. Покажем это на примере. Пусть имеется число 11001011, и маска 10100111, выполним сдвиг и побитовое сложение в разном порядке.

Исходное число X	11001011
Сдвиг влево $X \ll 1$	10010111
Маска M	10100111
Сумма по модулю два $(X \ll 1) \oplus M$	00110000

Исходное число X	11001011
Маска M	10100111
Сумма по модулю два $X \oplus M$	01101100
Сдвиг влево $(X \oplus M) \ll 1$	11011000

Результаты получились разные, это объясняется тем, что до сдвига и после сдвига сложение по модулю два для заданного бита маски происходит с разными битами числа.

Как вычислить маску, если нам известно исходное число и результат? Если бит числа не меняется (0 в исходном числе и 0 в результате, 1 в исходном числе и 1 в результате), то в маске соответствующий бит равен нулю. Если же бит меняется (с 0 на 1 или с 1 на 0), его надо сложить по модулю два с единицей, т.е. соответствующий бит маски равен единице. Таким образом, маска представляет собой побитовое сложение по модулю два результата с исходным числом. В этом можно убедиться на примерах выше.

Из этого следует, что, если инвертировать исходное число, маска тоже инвертируется. Пусть какой либо бит исходного числе равен соответствующему биту результата, тогда, как показано выше, соответствующий бит маски равен 0. После инверсии исходного числа его бит, наоборот, не равен соответствующему биту результата, следовательно, соответствующий бит маски должен быть равным 1. Пусть какой либо бит исходного числа не равен соответствующему биту результата, тогда соответствующий бит маски равен 1. После инверсии исходного числа его бит, наоборот, равен соответствующему биту результата, следовательно, соответствующий бит маски должен быть равным 0. Это продемонстрировано на примере ниже.

Исходное число (X)	11001011
Его инверсия ($\bar{X}$)	00110100
Маска (M)	10100111
Сумма по модулю два ($\bar{X} \oplus M$)	10010011

Исходное число (X)	11001011
Маска (M)	10100111
Инверсия маски ($\bar{M}$)	01011000
Сумма по модулю два ($X \oplus \bar{M}$)	10010011

Теперь подумаем, что происходит, если мы сначала складываем число с маской, а затем сдвигаем результат циклически влево? Например, мы сложили число X с маской M, получили число A, затем сдвинули его циклически влево, получили число B. Понятно, что A можно получить из B, если сдвинуть B циклически вправо. Ниже приведен пример.

Исходное число (X)	11001011
Маска (M)	10100111
Сумма по модулю два ($A = X \oplus M$)	01101100
Сдвиг влево ($B = A \ll 1$)	11011000

Таким образом, если имеется число X и результат R, у нас есть следующие возможности для поиска маски:

1. Вычисление маски как суммы по модулю два исходного числа с результатом, затем, возможно, инверсия маски ($M = X \oplus R$, $M = \overline{X \oplus R}$).

2. Сдвиг влево исходного числа, затем вычисление маски как суммы по модулю два исходного числа с результатом, затем, возможно, инверсия маски ($M = (X \ll 1) \oplus R$, $M = \overline{(X \ll 1) \oplus R}$).

3. Сдвиг вправо результата, зачем вычисление маски как суммы по модулю два числа с результатом, затем, возможно, инверсия маски $(M = X \oplus (R \gg 1)$, $M = \overline{X \oplus (R \gg 1)})$.

Нам остается подобрать маску, соответствующую заданным условиям.

Переведем сначала число и результат в двоичную систему счисления.

$$X = 81_{10} = 64 + 16 + 1 = 01010001_2$$

$$R = 55_{10} = 32 + 16 + 4 + 2 + 1 = 00110111_2$$

Способ 1.

X	01010001
R	00110111
$M = X \oplus R$	01100110
$M = \overline{X \oplus R}$	10011001

В полученных масках есть несколько нулей подряд.

Способ 2.

X	01010001
R	00110111
$X \ll 1$	10100010
$M = (X \ll 1) \oplus R$	10010101
$M = \overline{(X \ll 1) \oplus R}$	01101010

Маска $M = \overline{(X \ll 1) \oplus R}$ удовлетворяет заданным условиям: нет двух нулей, идущих подряд, но нули присутствуют .

Способ 3.

X	01010001
R	00110111
$R \gg 1$	10011011
$M = X \oplus (R \gg 1)$	11001010
$M = \overline{X \oplus (R \gg 1)}$	00110101

В полученных масках есть несколько нулей подряд.

Таким образом, алгоритм шифрования может быть таким:

1. Сдвиг влево;
2. Инверсия;
3. Сложение по модулю два с маской 01101010.

В следующей таблице показан результат проверки.

X	01010001
$X \ll 1$	10100010
$\overline{X \ll 1}$	01011101
M	01101010
$R = (\overline{X \ll 1}) \oplus M$	00110111

Алгоритм дешифрования состоит в следующем:

1. Число R в двоичной системе счисления складываем по модулю два с маской 00110111, получаем число $\overline{X \ll 1}$;
2. Число $\overline{X \ll 1}$ инвертируем, получаем число $X \ll 1$.
3. Число $X \ll 1$ циклически сдвигаем вправо, получаем число X.

Пример для того же числа R показан в таблице ниже.

R	00110111
M	01101010
$R \oplus M = \overline{X \ll 1}$	01011101
$X \ll 1$	10100010
X	01010001

Рассмотрим теперь вопрос, возможно ли найти такое число, которое не изменится после шифрования. В общем виде двоичное число X можно записать как $X = x_7x_6x_5x_4x_3x_2x_1x_0$. Здесь x_i – разряды числа, которые принимают значение 0 и 1. Примем во внимание, что для любого $z \in \{0,1\}$ верно $z \oplus 0 = z$, $z \oplus 1 = \bar{z}$ (эти соотношения легко проверить построением таблиц истинности). Посмотрим, что происходит с числом X в ходе шифрования.

X	$x_7x_6x_5x_4x_3x_2x_1x_0$
$X \ll 1$	$x_6x_5x_4x_3x_2x_1x_0x_7$
$\overline{X \ll 1}$	$\bar{x}_6\bar{x}_5\bar{x}_4\bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0\bar{x}_7$
M	0 1 1 0 1 0 1 0
$R = (\overline{X \ll 1}) \oplus M$	$\bar{x}_6x_5x_4\bar{x}_3x_2\bar{x}_1x_0\bar{x}_7$

Поскольку $R = X$, получаем уравнения для разрядов числа X

$$x_7 = \bar{x}_6; \quad x_3 = x_2;$$
$$x_6 = x_5; \quad x_2 = \bar{x}_1;$$
$$x_5 = x_4; \quad x_1 = x_0;$$
$$x_4 = \bar{x}_3; \quad x_0 = \bar{x}_7.$$

Зафиксировав, например, $x_7 = 1$, получаем

$$x_7 = \bar{x}_6 = 1; \quad x_3 = x_2 = 1;$$
$$x_6 = x_5 = 0; \quad x_2 = \bar{x}_1 = 1;$$
$$x_5 = x_4 = 0; \quad x_1 = x_0 = 0;$$
$$x_4 = \bar{x}_3 = 0; \quad x_0 = \bar{x}_7 = 0.$$

Система уравнений непротиворечива, т.е., имеет решение – это число 10001100 (140 в десятичной системе счисления. Проверка показана в таблице ниже.

X	10001100
$X \ll 1$	00011001
$\overline{X \ll 1}$	11100110
M	01101010
$R = (\overline{X \ll 1}) \oplus M$	10001100

Ответ. Алгоритм шифрования:

1. Циклический сдвиг влево;
2. Инверсия;
3. Сложение по модулю два с маской 01101010.

Алгоритм дешифрования:

1. Сложение по модулю два с маской 00110111;
2. Инверсия;
3. Циклический сдвиг вправо.

Число 140 (10001100 в двоичной системе) не меняется после шифрования.

Задачи в других вариантах аналогичны, поэтому приводится решение только для одного варианта.

Задание 7. Системы счисления

Задача 7. Природный газ проходит ряд промежуточных пунктов, прежде чем достигает конечного потребителя. После извлечения газа из скважины он по трубопроводу направляется на центральный пункт сбора, который находится на расстоянии 5670 (8) от скважины. Затем по большим трубам газ доходит до центрального распределительного узла, который находится на расстоянии 157C (16) от скважины. После газ проходит промежуточную компрессионную станцию, она находится на расстоянии 51004 (6) от центрального пункта сбора. Последней точкой перед потребителем становится распределительная станция, расстояние до которой в 2 раза больше чем расстояние от распределительного узла до промежуточной компрессионной станции.

Найти расстояние от скважины до распределительной станции и представить ответ в виде десятичного числа.

Примечание: Числа в скобках () – обозначают систему счисления, в которой записано число.

Решение задачи 7. Заменим для удобства все пункты буквами по порядку ABCDE:

A – скважина;

B – центральный пункт сбора;

C – центральный распределительный узел;

D – промежуточная компрессионная станция;

E – распределительная станция.

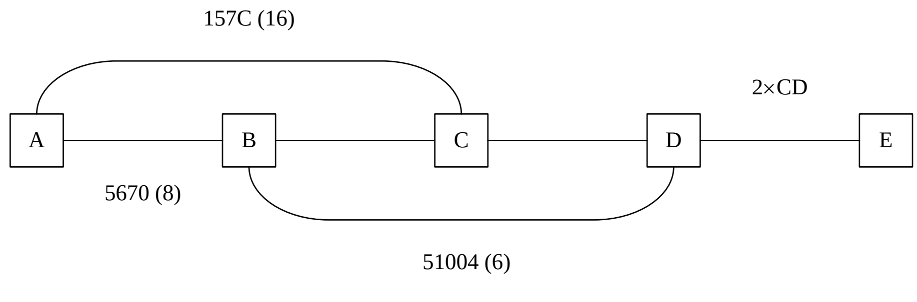
Рассмотрим расстояния:

От A до B − $5670_{(8)}$.

От A до C − $157C_{(16)}$.

От B до D − $51004_{(6)}$.

Расстояние от D до E − это число в 2 раза большее, чем C до D. Ниже показана общая схема.



Теперь приведем все числа к одной системе счисления. Ответ нужно дать в десятичной системе, поэтому в неё и будем переводить.

$AB=5670_{(8)} = 3000_{(10)}$

$AC=157C_{(16)} = 5500_{(10)}$

$BC=5500-3000=2500$

$BD=51004_{(6)} = 6700_{(10)}$

$CD=6700-2500=4200$

$DE=2\times4200 = 8400$

Общее расстояние: $AB+BC+CD+DE=3000+2500+4200+8400 = 18100$

Ответ:18100.

Задачи в остальных вариантах аналогичны, решение приводится только для одной.

Задание 8. Булева алгебра

Задача 8. На одной из расположенных далеко на севере нефтебаз случалась авария, связанная с подачей тепла в жилые помещения. Для того чтобы восстановить работоспособность нужно перезапустить всю систему. Выполнить перезапуск можно последовательно введёнными 8 комбинациями, которые состоят из положений вкл./выкл. для 4 рычагов (X, Y, Z, F). Вся сложность в том, что положение F зависит от положения других рычагов, мы смогли найти формулу для рычага F:

$$\boldsymbol{F} = ((\boldsymbol{X} \wedge \boldsymbol{Y}) \Rightarrow (\boldsymbol{X} \vee \boldsymbol{Y})) \wedge \overline{\boldsymbol{Z}}$$

Постройте таблицу истинности для формулы F

Решение задачи 8.

Необходимо определить порядок вычислений и построить таблицу истинности.

X	Y	Z	$\overline{Z}$	$(X \wedge Y)$	$(X \vee Y)$	$(X \wedge Y) \Rightarrow (X \vee Y)$	$(X \wedge Y) \Rightarrow (X \vee Y) \wedge \bar{Z}$
0	0	0	1	0	0	1	1
0	0	1	0	0	0	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	0

Задачи в остальных вариантах аналогичны, решение приводится только для одной.

Задание 9. Кодирование и декодирование

Задача 9. Доступ к пульту управления осуществляется с помощью пароля. Пароль автоматически сменяется каждый день и передается нам по закрытому каналу. Для большей безопасности было заранее оговорено что пароль передается в двоичной системе по 4 разряда (к примеру, 0000) на каждый символ, но, чтобы расшифровать пароль нужно добавить один разряд впереди, либо в конце. Только что поступил новый пароль, расшифруй его:

0111 1110 0011 0111 0001 0010

Решение задачи 9.

Нужно подставить ноль или единицу в начало или вконец, и записать все варианты.

Пример: Первая буква из 0111 имеет следующие варианты(**0**0111, **1**0111, 0111**1,** 0111**0**).

Получается двоичный пятиразрядный код, в котором, если букву **А** представить за 00000, то буква **Я** будет 11111. При этом буквы Ё и Е имеют одинаковый код, поскольку кодов всего 32. Слово из шести букв, для каждой буквы у нас есть 4 кандидата.

00111	01110	00011	00111	00001	00010	0 в начале
10111	11110	10011	10111	10001	10010	1 в начале
01110	11100	00110	01110	00010	00100	0 в конце
01111	11101	00111	01111	00011	00101	1 в конце

Заменяем коды буквами.

З	О	Г	З	Б	В
Ч	П	У	Ч	С	Т
П	Ъ	Ж	П	В	Д
О	Э	З	О	Г	Е

Составляем слово, выбирая по одной букве из каждого столбца. В итоге должно получиться слово, реально существующее в русском языке. Это слово – ПОГОСТ.

З	**О**	**Г**	З	Б	В
Ч	П	У	Ч	**С**	**Т**
П	Ъ	Ж	П	В	Д
О	Э	З	**О**	Г	Е

Ответ. ПОГОСТ (01111 01110 00011 01110 10001 10010).

Задачи в остальных вариантах аналогичны, решение приводится только для одной.

Задание 10. Количество информации

Задача 10. Руководство предприятия дало указ перевести действующую инструкцию в электронный вид и отправить на правки двум специалистам, которые бы внесли в нее необходимые изменения. Её изначальный размер после перепечатывания был 20 тыс. символов. Первый редактор уменьшил размер инструкции на 2 страницы. После редактирования вторым редактором её размер увеличился на 15% от начального объема.

Найти конечный размер инструкции в Кбайтах, округляя до целого в большую сторону, если 1 символ − 16 бит, а размер одной страницы −8800 бит

Решение задачи 10.

Необходимо выяснить, как документ изменился после каждого редактора.

Находим сколько символов на одной странице. На одной странице 8800/16 = 550 символов.

Первый редактор уменьшил на 2 страницы. 2 страницы это 2×550 = 1100 символов.

Второй редактор увеличил объем инструкции на 15% от начального объема (20000 символов), значит берём 15% от 20 000, это 3000 символов.

Считаем окончательный объем: 20 000(это начальное число) + 3000(второй редактор увеличил) – 1100(первый редактор уменьшил) = 21900 символов.

Один символ равен 16 бит, это 2 байта.

Перевод символов в байты 21900×2 = 43800 байт.

Перевод в Кбайт: 43800 байт / (1024) = 42,7734 Кбайт.

Ответ округляем до целого в большую сторону.

Ответ: 43 Кбайт.

Задачи в остальных вариантах аналогичны, решение приводится только для одной.