ИНТЕЛЛЕКТУАЛЬНЫЕ РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ

Профиль “Интеллектуальные робототехнические системы” посвящен решению классических задач робототехники: автономной локализации мобильного наземного робототехнического устройства, планирования и построения маршрута мобильного робототехнического устройства или его частей, навигации, распознавания графической информации.

Профиль включает в себя задачи по двум школьным предметам: **математика** и **информатика**.

§1 Первый отборочный этап

Первый отборочный тур проводится индивидуально в сети Интернет, работы оцениваются автоматически средствами системы онлайн-тестирования. Для каждого из параллелей (9 класс или 10-11 класс) предлагается свой набор задач по математике, задачи по информатике - общие для всех участников. Решение задач по информатике предполагало написание программ, допускалось использовать один из нескольких языков программирования: C, C++, Java, Python или JavaScript. На решение задач каждого предмета первого отборочного этапа участникам давалось 2 дня. У участников было два временных слота по 2 дня каждый, когда они могли решать задачи по предмету. Решение каждой задачи дает определенное количество баллов. Баллы зачисляются в полном объеме за правильное решение задачи. Участники получают оценку за решение задач в совокупности по всем предметам данного профиля (математика и информатика) — суммарно от 0 до 20 баллов.

1.1 Первая попытка

Задачи по математике (9 класс)

Задача 1.1.1 (1 балл)

Условие:

Известно, что $a^{16} = 2$. Вычислите сумму

$$\frac{1}{1-a}+\frac{1}{1+a}+\frac{2}{1+a^2}+\frac{4}{1+a^4}+\frac{8}{1+a^8}$$

Решение:

Преобразуем сумму

$$S=\left(\frac{1}{1-a}+\frac{1}{1+a}\right)+\frac{2}{1+a^2}+\frac{4}{1+a^4}+\frac{8}{1+a^8}=$$

$$=\left(\frac{2}{1-a^2}+\frac{2}{1+a^2}\right)+\frac{4}{1+a^4}+\frac{8}{1+a^8}=\left(\frac{4}{1-a^4}+\frac{4}{1+a^4}\right)+\frac{8}{1+a^8}=$$

$$=\left(\frac{8}{1-a^8}+\frac{8}{1+a^8}\right)=\frac{16}{1-a^{16}}$$

Теперь легко найти чему равна сумма при известном a:

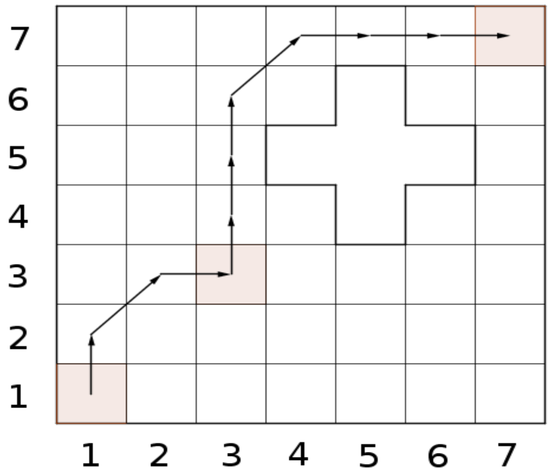
$$S=\frac{16}{1-a^{16}}=\frac{16}{1-2}=-16$$

Ответ: -16.

Задача 1.1.2 (2 балла)

Условие:

Приведенное справа поле состоит из 44 единичных клеток. Каждым своим ходом шахматный король может пойти либо на одну клетку вверх, либо на одну клетку вправо, либо на одну клетку вверх-вправо по диагонали (не выходя при этом за границы поля). Сколько существует путей короля, ведущих из клетки (1, 1) в клетку (7, 7), проходящих через клетку (3, 3) (на рисунке изображен один из возможных путей)?



Решение:

В каждую клетку нашего поля напишем число путей, ведущих в нее из клетки (1, 1), удовлетворяющих условию задачи. Очередное число в клетке вычисляется как сумма чисел в клетках непосредственно слева, снизу и слева-снизу. В итоге мы получаем, что число путей, ведущих в клетку (7, 7) равно 156 (см. рисунок).

		13	52	78	78	156
		13	26		0	78
		13				78
		13	39		26	52
1	5	13	13	13	13	13
1	3	5				
1	1	1				

Ответ: 156

Задание 1.1.3 (2 балла)

Условие:

Найдите значение выражения $\alpha^6+\beta^6$, где α и β -- корни квадратного трехчлена x^2+5x+3.

Решение:

Заметим, что

$$\alpha^6+\beta^6=(\alpha^3+\beta^3)^2-2(\alpha\cdot\beta)^3=\left((\alpha+\beta)^3-3(\alpha+\beta)\cdot(\alpha\cdot\beta)\right)^2-2(\alpha\cdot\beta)^3$$

Осталось вспомнить теорему Виета для квадратного трехчлена
$x^2 + 5x + 3$

$$\begin{cases} \alpha + \beta = -\dfrac{5}{1} = -5 \\ \alpha \cdot \beta = \dfrac{3}{1} = 3 \end{cases}$$

Подставим полученные значения в наше выражение

$$\alpha^6 + \beta^6 = \left((-5)^3 - 3 \cdot (-5) \cdot 3\right)^2 - 2 \cdot 3^3 = 6346$$

Ответ: 6346

Задача 1.1.4 (2 балла)

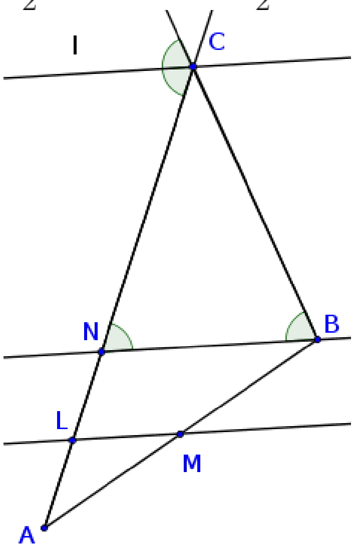
Условие:

Пусть l - биссектриса внешнего угла C треугольника ABC. Прямая, параллельная l и проходящая через середину M стороны AB, пересекает AC в точке L. Найти длину отрезка A, если AC = 5,3 и CB = 3,2.

Решение:

Проведем через точку B прямую параллельную l, точку пересечения с отрезком AC обозначим за N (см. рисунок). Тогда треугольник NCB равнобедренный (CN = CB). Так как ML || BN и M - середина отрезка AB, то ML -- средняя линия треугольника BAN, следовательно NL = LA. В итоге получаем

$$AL = \frac{1}{2}NA = \frac{1}{2}(CA - CN) = \frac{1}{2}(CA - CB) = 1,05$$



Ответ: 1,05

Задача 1.1.5 (3 балла)

Условие:

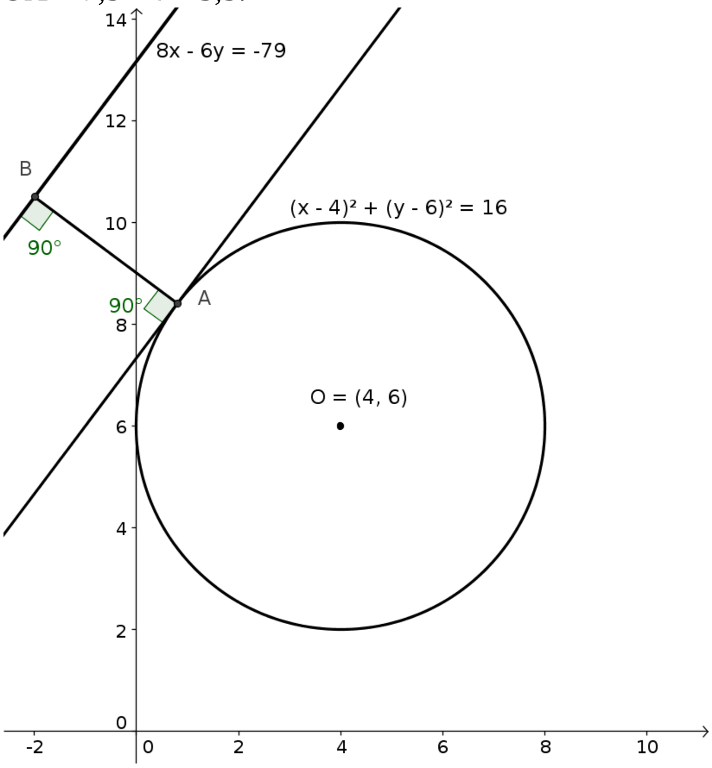
Точка A лежит на окружности $x^2 + y^2 - 8x - 12y + 36 = 0$, а точка B -- на прямой $-8x + 6y - 79 = 0$. Найдите наименьшее возможное расстояние между точками A и B.

Решение:

Уравнение окружности преобразуем к виду $(x-4)^2 + (y-6)^2 = 4^2$. Получаем окружность с центром в точке O(4, 6) радиуса 4. Длина отрезка AB будет минимальна, когда она совпадает с расстоянием между параллельными прямыми, одна из которых исходная, а вторая касается окружности (см. рисунок). Это расстояние можно найти как разность расстояний от точки O до наших параллельных прямых AB = OB - OA (точки O, А и В лежат на одной прямой). Расстояние OA = r = 4, а расстояние OB можно вычислить как расстояние от точки до прямой

$$OB = \frac{|8 \cdot 4 + (-6) \cdot 6 + 79|}{\sqrt{8^2 + (-6)^2}} = 7,5$$

В итоге AB = OB - OA = 7,5 - 4 = 3,5.



Ответ: 3,5

1.2 Первая попытка
Задачи по математике (10-11 класс)

Задача 1.2.1 (1 балл)

Условие:

Известно, что $a = \sqrt[2048]{2}$. Вычислите сумму

$$\frac{1}{1-a} + \frac{1}{1+a} + \frac{2}{1+a^2} + \cdots + \frac{512}{1+a^{512}} + \frac{1024}{1+a^{1024}}$$

Решение:

Преобразуем сумму

$$S(a) = \left(\frac{1}{1-a} + \frac{1}{1+a}\right) + \frac{2}{1+a^2} + \cdots + \frac{1024}{1+a^{1024}} =$$

$$= \left(\frac{2}{1-a^2} + \frac{2}{1+a^2}\right) + \frac{4}{1+a^4} + \cdots + \frac{1024}{1+a^{1024}} = \cdots =$$

$$= \left(\frac{1024}{1-a^{1024}} + \frac{1024}{1+a^{1024}}\right) = \frac{2048}{1-a^{2048}}$$

Теперь легко найти, чему равна сумма при $a = \sqrt[2048]{2}$:

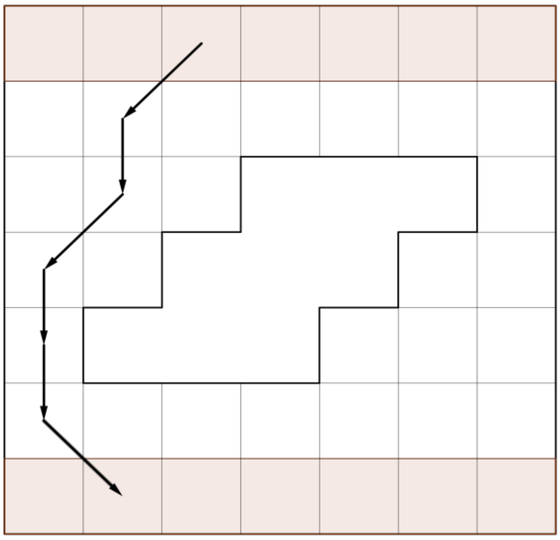
$$S(\sqrt[2048]{2}) = \frac{2048}{1-(\sqrt[2048]{2})^{2048}} = -2048$$

Ответ: -2048

Задача 1.2.2 (2 балла)

Условие:

Приведенное справа поле состоит из 40 единичных клеток. Кузнечик из каждой клетки может прыгнуть либо на одну клетку вниз, либо на одну клетку влево-вниз по диагонали, либо на одну клетку вправо-вниз по диагонали (не выходя при этом за границы поля). Сколько существует путей кузнечика, ведущих из верхнего ряда квадратов в нижний? (На рисунке изображен один из возможных путей, верхний и нижний ряды квадратов выделены сиреневым цветом.)



Решение:

В каждую клетку нашего поля напишем число путей кузнечика, ведущих в нее из верхнего ряда клеток. Очередное число в клетке вычисляется как сумма чисел в клетках непосредственно слева-сверху, сверху и справа-сверху. В итоге мы получаем, что число путей кузнечика, ведущих в клетки нижнего ряда равно сумме 70 + 70 + 40 + 20 + 45 + 60 + 45 = 350 (см. рисунок).

1	1	1	1	1	1	1
2	3	3	3	3	3	2
5	8	9				5
13	22				5	5
35				5	10	10
35	35	0	5	15	25	20
70	70	40	20	45	60	45

Ответ: 350

Задача 1.2.3 (2 балла)

Условие:

Найдите значение выражения $\alpha^3+\beta^3+\gamma^3$, где α, β и γ - корни кубического многочлена x^3+x^2-7x+4.

Решение:

Заметим, что

$$\alpha^3+\beta^3+\gamma^3=(\alpha+\beta+\gamma)^3-3(\alpha+\beta+\gamma)(\alpha\beta+\beta\gamma+\gamma\alpha)+3\alpha\beta\gamma$$

Осталось вспомнить теорему Виета для многочлена x^3+x^2-7x+4

$$\begin{cases} \alpha+\beta+\gamma=-\dfrac{1}{1}=-1 \\ \alpha\beta+\beta\gamma+\gamma\alpha=\dfrac{-7}{1}=-7 \\ \alpha\beta\gamma=-\dfrac{4}{1}=-4 \end{cases}$$

Подставим полученные значения в наше выражение

$$\alpha^3+\beta^3+\gamma^3=(-1)^3-3(-1)(-7)+3(-4)=-34$$

Ответ: -34

Задача 1.2.4 (2 балла)

Условие:

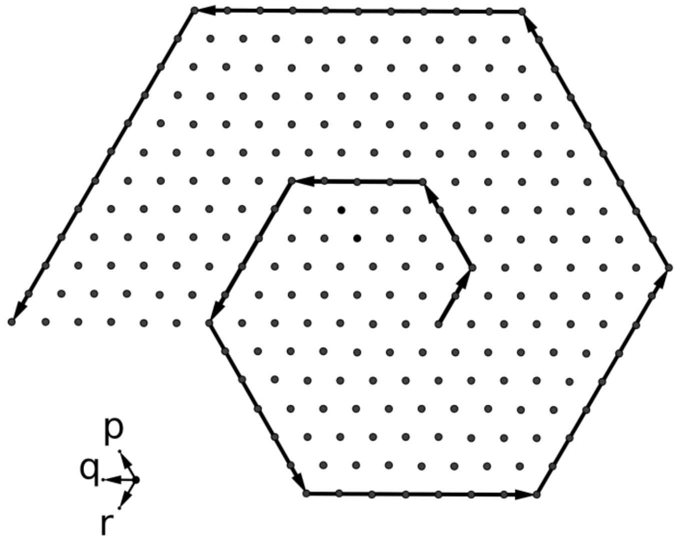
Саша собрал робота, который едет со скоростью 10 м/мин и запустил его на ровной открытой площадке. Робот проехал 2 минуты в одном направлении и повернул налево на 60 градусов, затем 3 минуты проехал прямо и повернул налево на 60 градусов, потом проехал 4 минуты и снова повернул налево на 60 градусов и т.д. после каждого поворота до следующего поворота двигался на минуту дольше. Через 65 минут закончился заряд батареи и робот остановился. На каком расстоянии робот оказался от начальной точки?

Решение:

Так как через 65 минут закончится заряд батареи, то робот проедет в итоге 20 м, 30 м, ..., 110 м при этом каждый раз поворачивая налево на 60 градусов (см. рисунок, расстояние между соседними точками 10 м). Всю траекторию робота можно представить в виде линейной комбинации трех векторов $\vec{p},\vec{q}$ и $\vec{r}$ (длины равны 10 м). В итоге робот удалится от начальной точки на вектор

$$\vec{l}=-2\vec{r}+3\vec{p}+4\vec{q}+5\vec{r}-6\vec{p}-7\vec{q}-8\vec{r}+9\vec{p}+10\vec{q}+11\vec{r}$$

Сокращая, получаем $\vec{l}=6\vec{p}+7\vec{q}+6\vec{r}$. А так как $\vec{p}+\vec{q}=\vec{r}$, то $\vec{l}=6(\vec{p}+\vec{q})+7\vec{r}=13\vec{r}$. В итоге получаем, что длина вектора $|\vec{l}|=13|\vec{r}|=130$ м.



Ответ: 130 м.

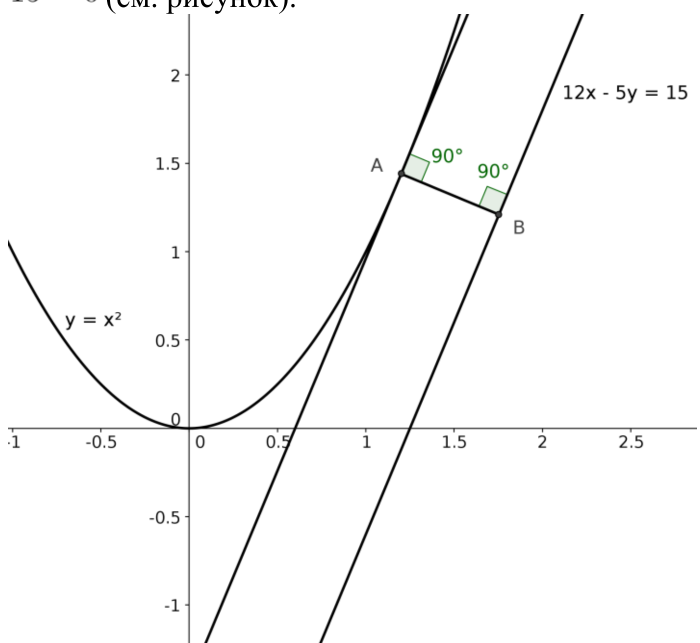
Задача 1.2.5 (3 балла)

Условие:

Точка A лежит на параболе $y = x^2$, а точка B - на прямой $12x - 5y - 15 = 0$. Найдите наименьшее возможное расстояние между A и B.

Решение:

Для того, чтобы найти минимальное расстояние от точки A до точки B нужно провести прямую параллельную $12x - 5y - 15 = 0$, которая будет касаться параболы. A - точка касания этой прямой и параболы, B - перпендикуляр, опущенный из точки A на прямую $12x - 5y - 15 = 0$ (см. рисунок).



Уравнение исходной прямой можно записать $y = 2,4x - 3$, следовательно коэффициент k = 2,4. Мы получаем, что уравнение касающейся прямой имеет вид $y = 2,4x - a$. Так как точка пересечения с параболой $y = x^2$ единственная, то квадратное уравнение $x^2 = 2,4x - a$ имеет дискриминант равный 0, следовательно а = 1,44. Длину отрезка АВ можно найти как расстояние между параллельными прямыми

$$AB = \frac{|c_1 - c_2|}{\sqrt{k^2+1}} = \frac{|-3-(-1,44)|}{\sqrt{2,4^2+1}} = 0,6$$

Ответ: 0,6

1.3. Первая попытка
Задачи по информатике

Задача 1.3.1 (1 балл)

Условие:

Складской робот-погрузчик находится внутри одного из помещений в большом логическом центре. Проекция помещения — прямоугольник, координаты противоположных углов которого находятся в точках (0, 0) и (n, m). Вдоль стенок расположены грузы, которые погрузчик должен загрузить в себя. Какое минимальное расстояние пройдет погрузчик, чтобы добраться до одной из стенок складского помещения?

Формат входных данных:

Первая строка содержит n, m $(1 \le n, m \le 10^9)$, ширину и длину складского помещения. Вторая строка содержит два числа x, y $(0 \le x \le n, 0 \le y \le m)$, координаты погрузчика.

Формат выходных данных:

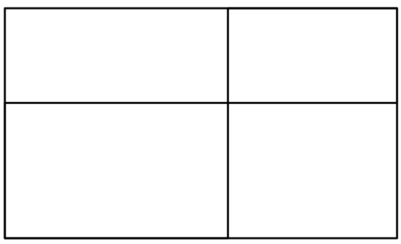
Выведите одно целое число - минимальное расстояние до одной из стенок.

Пример:

stdin:
3 4
1 2
***stdout*:**
1

Решение:

Требуется найти минимум среди четырех сторон.



Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    n, m, x, y = [int(t) for t in dataset.split()]
    return str(min(x, y, n - x, m - y))

print(solve(sys.stdin.read()))
```

Задача 1.3.2 (2 балла)

Условие:

В логистическом центре 5 роботов-погрузчиков занимались погрузкой ящиков с грузами. Первый робот перетащил ровно A% от всех ящиков, второй робот — ровно B% от всех ящиков, третий робот — ровно C%, четвертый робот — ровно D%, пятый робот перетащил ровно E%.

Какое минимальное количество ящиков было погружено роботами?

Формат входных данных:

Единственная строка входных данных содержит пять чисел A, B, C, D, E, $(A + B + C + D + E = 100, 0 \le A, B, C, D, E \le 100)$

Формат выходных данных:

Выведите минимальное количество погруженных ящиков.

Пример:

stdin:
15 55 0 5 25
stdout:
20

Решение:

Очевидно, что минимальное количество ящиков не превысит 100, это просто A+B+C+D+E. То есть достаточно перебрать возможные ответы и проверить их на правильность, среди них выбрать минимальный.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    a = [int(x) for x in dataset.split()]
    for i in range(1, 101):
        failed = False
        for x in a:
            if i * x % 100 != 0:
                failed = True
        if not failed:
            return str(i)
    return str(100);

print(solve(sys.stdin.read()))
```

Задача 1.3.3 (2 балла)

Условие:

Каждая задача для робота на погрузку на логистическом заводе определяется координационной системой. Система для определения номера робота использует 3 числа, присвоенные каждому грузу: идентификатор получателя, идентификатор грузовой компании и идентификатор отправителя. Номер робота определяется путем перевода перечисленных идентификаторов в двоичную систему счисления и побитовым применением булевой функции от трех аргументов:

X	***Y***	***Z***	***f(x, y, z)***
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

где X — это n-ый бит из идентификатора получателя, Y — n-ый бит из идентификатора грузовой компании, а Z — n-ый бит из идентификатора отправителя.

Найдите номер робота путем применения этой функции к идентификаторам груза.

Формат входных данных:

Единственная строка входных данных содержит три числа x, y, z $(0 \leq x, y, z \leq 2^{64} - 1)$ — идентификатор получателя, идентификатор грузовой компании и идентификатор отправителя, соответственно.

Формат выходных данных:

Выведите номер требуемого робота.

Пример №1:

stdin:
76 58 109
stdout:
40

Пример №2:

stdin:
7 8 5
stdout:
2

Пояснение:

Побитовое применение функции означает, что функция применяется к каждому разряду числа в двоичном представлении.

Пример:
$x = 76_{10} = 1001100_2$

$y = 58_{10} = 111010_2$
$z = 109_{10} = 1101101_2$

Применяем функцию начиная с младших разрядов, при нехватке дополняем лидирующими нулями.

$ans = 0101000_2 = 40_{10}$

Решение:

Задача на реализацию, требуется просто провести битовые манипуляции с числами.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    a, b, c = ["{:064b}".format(int(t))
               for t in dataset.split()]
    ans = [0] * 64;
    for i in range(0, 64):
        if a[i] == '1' and b[i] == '1':
            ans[i] = 1;
        if b[i] == '1' and c[i] == '1':
            ans[i] = 1;
        if a[i] == '1' and b[i] == '0' and c[i] == '0':
            ans[i] = 1;
    s = ''.join(str(x) for x in ans)
    return str(int(s, 2))

print(solve(sys.stdin.read()))
```

Задача 1.3.4 (2 балла)

Условие:

На логистическом центре каждый ящик маркируется парой символов латинского алфавита, при этом используются только строчные буквы (пример маркировки, ab или bc). Для отправки грузов из логистического центра, грузы должны быть упакованы по 3 ящика. Для того, чтобы грузы были правильно упакованы по три действует следующее правило: три ящика можно упаковать если первая буква маркировки среднего ящика совпадает с последней буквой первого ящика, а последняя буква среднего ящика совпадает с первой буквой третьего (например, ab|bc|cd — упакованы верно).

Необходимо узнать, сколько различных упаковок можно составить из заданного набора ящиков с грузами, если каждый ящик можно использовать в упаковке только один раз. Две упаковки s и t считаются различными, если существует такой индекс i, что маркировка s_i не равна маркировке t_i.

Формат входных данных:

Первая строка входных данных содержит целое число n — количество ящиков с грузами $(3 \leq n \leq 100)$. Во второй строке записаны n маркировок грузов, разделенных пробелом.

Формат выходных данных:

Выведите количество различных упаковок.

Пример №1:

stdin:

3
ab bc cd
stdout:
1

Пример №2:

stdin:
3
ab bb ba
stdout:
3

Пример №3:

stdin:
5
ab bc da ca bd
stdout:
8

Решение:

Давайте перебирать тройки грузов. Остается только убрать повторяющиеся тройки, для этого можно использовать структуры данных set/hashset, либо добавив все тройки в массив и отсортировав их. После чего можно за линейное время найти количество уникальных. Сложность такого алгоритма $O\left(n^3 \cdot \ln\left(n^3\right)\right)$.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    a = dataset.split()
    n = int(a[0])
    a = a[1:]
    s = set()
    for i in range(n):
        for j in range(n):
            for k in range(n):
                if i != j and j != k and i != k and
                   a[i][1] == a[j][0] and
                   a[j][1] == a[k][0]:
                        s.add(a[i]+a[j][1]+a[k][1])
    return  str(len(s))

print(solve(sys.stdin.read()))
```

Задача 1.3.5 (3 балла)

Условие:

В логистическом центре у одного из роботов-погрузчиков обнаружилась проблема в программном обеспечении. Чтобы обнаружить сбойного робота персонал центра собрался в течение ночи произвести диагностику всех роботов. Для этого они арендуют диагностические стенды у компании-разработчика роботов.

Один стенд может делать диагностику для любого количества роботов, при этом один и тот же робот может подключаться на диагностику к нескольким стендам. Каждый стенд выдает результат диагностики только утром, после проверки всех роботов, которые были подключены к нему, но есть известная проблема стенда – он может сказать, что результат диагностики роботов отрицательный, но у какого конкретно робота – неизвестно.

Стоимость аренды диагностических стендов очень высокая, поэтому задача руководства центра арендовать как можно меньше таких стендов. Помоги логистическому центру определить, какое минимальное количество диагностических стендов нужно для тестирования роботов, чтобы утром они уже знали проблемного робота.

Формат входных данных:

Единственная строка входных данных содержит число x $(1 \leq x \leq 10^9)$ — количество роботов.

Формат выходных данных:

Выведите минимальное количество стендов.

Пример №1:

stdin:

7

stdout:

3

Пример №2:

stdin:

2

stdout:

1

Решение:

Оптимальная стратегия: выпишем номера роботов в двоичной системе счисления.

$$1_{10} = 1_2, 2_{10} = 10_2, 3_{10} = 11_2, 4_{10} = 100_2 \ldots$$

Подключим к первому стенду тех роботов, у которых первый бит равен одному, ко второму стенду - роботов, у которых второй бит равен одному, к третьему - третий и так далее. На следующий день смотрим, на каких стендах результат диагностики роботов отрицательный. Если робот - сбойный, значит все стенды, к каким он был подключен, сигнализируют об отрицательной диагностике. По номерам стендов можно восстановить номер робота (перевод из двоичной системы в десятичную). При этом последнего робота можно не проверять, так как если он - сбойный, то ни один стенд не сигнализирует об этом. То есть для восьми роботов требуется три стенда, а не четыре.

Ответ на задачу: $\log_2(n)$ с округлением вверх.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    n = int(dataset.split()[0])
    if n == 1:
        return "0"
    n -= 1
    return str(len("{:b}".format(n)))
```

```
print(solve(sys.stdin.read()))
```

1.4 Вторая попытка

Задачи по математике (9 класс)

Задача 1.4.1 (1 балл)

Условие:

Найдите остаток при делении $2016 \cdot 2017 \cdot 2018 \cdot 2019 \cdot 2020$ на 11.

Решение:

Так как

$$2016 \equiv 3 \pmod{11},\ 2017 \equiv 4 \pmod{11}, \ldots,\ 2020 \equiv 7 \pmod{11},$$

то

$$2016 \cdot 2017 \cdot 2018 \cdot 2019 \cdot 2020 \equiv 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \equiv 2520 \equiv 1 \pmod{11}$$

Ответ: 1

Задача 1.4.2 (2 балла)

Условие:

В классе из 16 учащихся только 2 сделали домашнее задание. Равиль Ниязович наугад выбирает школьника для ответа. Если задание не сделано, то учащийся получает оценку “два” и садится. Затем процесс продолжается до тех пор, пока не ответит школьник, сделавший домашнее задание. Найдите вероятность того, что Равиль Ниязович поставит ровно три двойки. (Если ответом является бесконечная дробь, то результат округлите до сотых.)

Решение:

Наше вероятностное пространство состоит из 16! исходов (число всех перестановок на 16-ти учениках). Посчитаем в скольких из них Равиль Ниязович поставит ровно три двойки: первого опрашиваемого школьника можно выбрать 14 способами, второго - 13, третьего - 12, четвертого - 2, а остальных 12 школьников можно переставлять как угодно. Всего подходящих исходов будет $14 \cdot 13 \cdot 12 \cdot 2 \cdot 12!$. Следовательно вероятность вычисляется

$$P = \frac{14 \cdot 13 \cdot 12 \cdot 2 \cdot 12!}{16!} = \frac{1}{10} = 0,1$$

Ответ: 0,1

Задача 1.4.3 (2 балла)

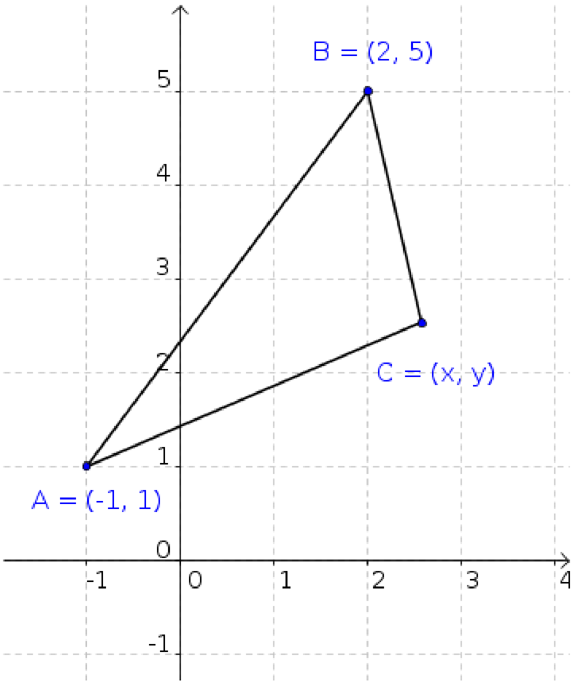
Условие:

Найдите минимальное значение выражения

$$\sqrt{(x+1)^2 + (y-1)^2} + \sqrt{(x-2)^2 + (y-5)^2}$$

Решение:

Отметим на координатной плоскости три точки A(-1, 1), B(2, 5) и C(x, y),



тогда

$$AC = \sqrt{(x+1)^2 + (y-1)^2}$$

и

$$BC = \sqrt{(x-2)^2 + (y-5)^2}$$

Следовательно, наше выражение равно сумме расстояний от точки C до точек A и B. По неравенству треугольника

$$AC + BC \geq AB = \sqrt{(2-(-1))^2 + (5-1)^2} = 5$$

Причем равенство достигается, когда точка C лежит на отрезке AB.

Ответ: 5

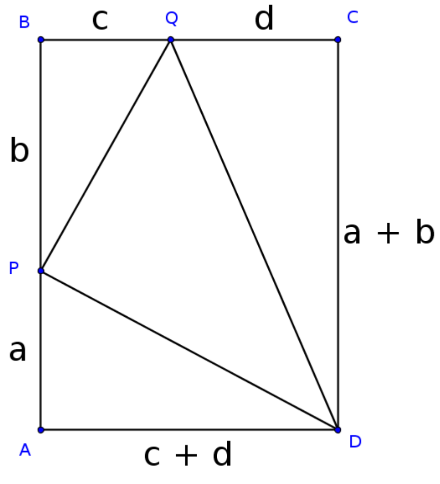
Задача 1.4.4 (2 балла)

Условие:

На сторонах AB и BC прямоугольника ABCD отмечены точки P и Q соответственно. Оказалось, что $S_{ABCD} = 2017$ и $S_{DPQ} = 505$. Чему может быть равно произведение $AP \cdot CQ$?

Решение:

Обозначим длины отрезков следующим образом AP = a, PB = b, BQ = c и QC = d.



Тогда

$$S_{ABCD} = (a+b)\cdot(c+d) = a\cdot c + a\cdot d + b\cdot c + b\cdot d$$

и

$$S_{DPQ} = S_{ABCD} - S_{PDA} - S_{PQB} - S_{QCD} =$$
$$= (a+b)\cdot(c+d) - \frac{a\cdot(c+d)}{2} - \frac{b\cdot c}{2} - \frac{d\cdot(a+b)}{2} =$$
$$= \frac{a\cdot c + b\cdot c + b\cdot d}{2}$$

А так как AP · CQ = a · d, то получаем

$$AP \cdot CQ = S_{ABCD} - 2S_{PDQ} = 2017 - 2\cdot 505 = 1007$$

Ответ: 1007

Задача 1.4.5 (3 балла)

Условия:

Даны две последовательности чисел (x_n) и (y_n) следующими условиями: $x_0 = 3-\sqrt{2}$, $y_0 = 3+\sqrt{2}$ и $\forall n \in \cup\{0\}$

$$x_{n+1} = \frac{x_n + y_n}{\sqrt{2}},\ y_{n+1} = \frac{y_n - x_n}{\sqrt{2}}.$$

Найдите $x_{2016} + y_{2016}$.

Решение:

Заметим, что

$$x_{n+1} + y_{n+1} = \sqrt{2}y_n \text{ и } y_{n+1} - x_{n+1} = -\sqrt{2}x_n$$

Поэтому

$$x_{n+2} = \frac{x_{n+1} + y_{n+1}}{\sqrt{2}} = y_n \text{ и } y_{n+2} = \frac{y_{n+1} - x_{n+1}}{\sqrt{2}} = -x_n$$

В итоге

$$(x_{n+8}, y_{n+8}) = (y_{n+6}, -x_{n+6}) = (-x_{n+4}, -y_{n+4}) = (-y_{n+2}, x_{n+2}) = (x_n, y_n).$$

Обе наши последовательности имеют период равный 8, а так как 2016 = 252 · 8, то $(x_{2016}, y_{2016}) = (x_0, y_0)$. Следовательно,

$$x_{2016} + y_{2016} = x_0 + y_0 = (3-\sqrt{2}) + (3+\sqrt{2}) = 6$$

Ответ: 6

1.5 Вторая попытка

Задачи по математике (10-11 класс)

Задача 1.5.1 (1 балл)

Условие:

Найдите остаток при делении 9^{2017} на 11.

Решение:

Так как $9 \equiv -2 \pmod{11}$, то

$$9^{2017} \equiv (-2)^{2017} \equiv -2^{2017} \pmod{11}$$

Осталось заметить, что $2^{2017} = 4\cdot(2^5)^{403}$ $2^5 \equiv -1 \pmod{11}$. Поэтому

$$9^{2017} \equiv -2^{2017} \equiv -4\cdot(-1)^{403} \equiv 4 \pmod{11}$$

Ответ: 4

Задача 1.5.2 (2 балла)

Условие:

Найдите значение выражения

$$3\cdot\cos\left(\arcsin\frac{1}{2}\right)\cdot\cos\left(\arcsin\frac{1}{3}\right)\cdot\ldots\cdot\cos\left(\arcsin\frac{1}{288}\right)$$

Решение:

Так как

$$\cos\left(\arcsin\frac{1}{n}\right)=\sqrt{1-\frac{1}{n^2}}=\frac{\sqrt{(n-1)(n+1)}}{n}$$

получаем

$$S=3\cdot\cos\left(\arcsin\frac{1}{2}\right)\cdot\cos\left(\arcsin\frac{1}{3}\right)\cdot\ldots\cdot\cos\left(\arcsin\frac{1}{288}\right)=$$
$$=3\cdot\frac{\sqrt{1\cdot 3}}{2}\cdot\frac{\sqrt{2\cdot 4}}{3}\cdot\ldots\cdot\frac{\sqrt{287\cdot 289}}{288}$$

Каждое натуральное число от 3 до 287 встретится ровно 2 раза под корнем в числителе, поэтому эти числа сокращаются с числами, стоящими в знаменателе.

$$S=3\cdot\frac{\sqrt{1\cdot 2\cdot 288\cdot 289}}{2\cdot 288}=3\cdot\sqrt{\frac{289}{576}}=\frac{17}{8}=2,125$$

Ответ: 2,125

Задача 1.5.3 (2 балла)

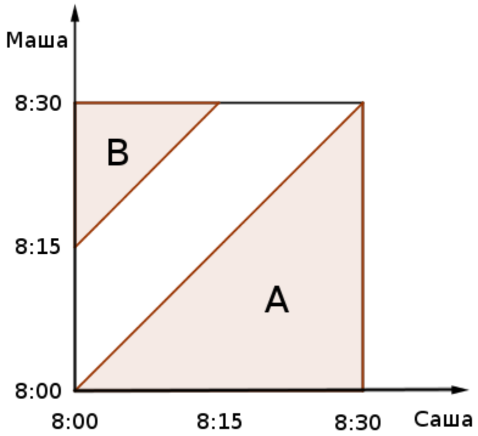
Условие:

Прием у терапевта начинается в 8 утра и для каждого пациента длится 15 минут. Саша и Маша записались на прием с 8.00 до 8.30 и приходят в случайное время в данном промежутке независимо друг от друга. Других пациентов в это время нет. Какова вероятность того, что Маше не придется ждать своей очереди?

Решение:

Рассмотрим конфигурационное пространство времен приходов Саши и Маши. Для того, чтобы Маше не пришлось ждать своей очереди, ей нужно либо прийти раньше Саши, либо на 15 или более минут позже. Первому случаю соответствуют точки, лежащие в области A конфигурационного пространства (см. рисунок), а второму случаю -- точки, лежащие в области B. Тогда ответ на эту задачу

$$P(A)+P(B)=\frac{1}{2}+\frac{1}{8}=0,625$$



Ответ: 0,625

Задача 1.5.4 (2 балла)

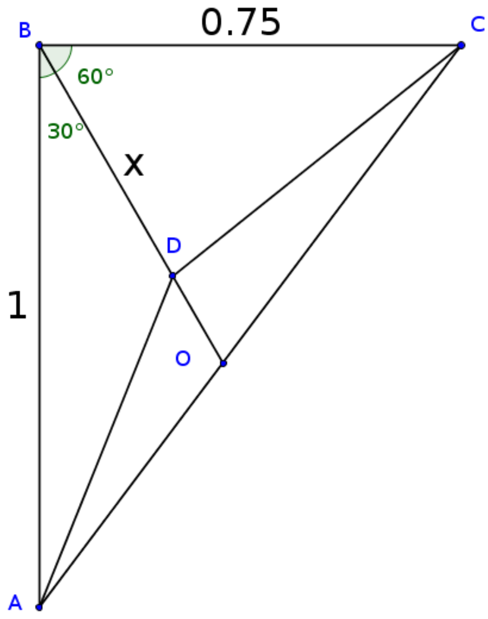
Условие:

Найдите минимум функции

$$f(x) = \sqrt{\frac{9}{16} - \frac{3}{4}x + x^2} + \sqrt{1 - \sqrt{3}x + x^2}$$

Решение:

При решении этой задачи будем использовать теорему косинусов. Рассмотрим прямоугольный треугольник ABC и отложим точку D как показано на рисунке.



Тогда по теореме косинусов для треугольника ABD:

$$AD^2 = AB^2 + BD^2 - 2AB \cdot BD \cdot \cos 30° = 1 - \sqrt{3}x + x^2$$

и по теореме косинусов для треугольника CBD:

$$CD^2 = CB^2 + BD^2 - 2CB \cdot BD \cdot \cos 60° = \frac{9}{16} - \frac{3}{4}x + x^2$$

Получаем, что значение нашей функции

$$f(x) = AD + CD$$

По неравенству треугольника

$$AD + CD \geq AC = \sqrt{1^2 + 0,75^2} = 1,25$$

причем равенство достигается, когда точка D лежит на гипотенузе AC, то есть совпадает с точкой O (см. рисунок).

Ответ: 1,25

Задача 1.5.5 (3 балла)

Условие:

Даны две последовательности чисел (x_n) и (y_n) следующими условиями: $x_0 = 1 - \sqrt{2}$, $y_0 = 1 + \sqrt{2}$ и $\forall n \in \cup\{0\}$

$$x_{n+1} = x_n \cdot \cos 5^\circ + y_n \cdot \sin 5^\circ, y_{n+1} = y_n \cdot \cos 5^\circ - x_n \cdot \sin 5^\circ.$$

Найдите $x_{2016} + y_{2016}$.

Решение:

Рассмотрим вектора $\vec{v}_n = (x_n, y_n)$. Заметим, что

$$|\vec{v}_{n+1}|^2 = x_{n+1}^2 + y_{n+1}^2 = (x_n \cdot \cos 5^\circ + y_n \cdot \sin 5^\circ)^2 + (y_n \cdot \cos 5^\circ - x_n \cdot \sin 5^\circ)^2 =$$
$$= x_n^2 \cdot (\cos^2 5^\circ + \sin^2 5^\circ) + y_n^2 \cdot (\cos^2 5^\circ + \sin^2 5^\circ) = x_n^2 + y_n^2 = |\vec{v}_n|^2$$

Следовательно, длины векторов не изменяются

$$|\vec{v}_{n+1}| = |\vec{v}_n| = \ldots = |\vec{v}_0| = \sqrt{(1-\sqrt{2})^2 + (1+\sqrt{2})^2} = \sqrt{6}$$

Пусть α_n ориентированный угол между лучом Ox и вектором $\vec{v}_n$. Тогда

$$x_n = \sqrt{6} \cdot \cos \alpha_n, y_n = \sqrt{6} \cdot \sin \alpha_n$$

следовательно,

$$\cos \alpha_{n+1} = \cos \alpha_n \cdot \cos 5^\circ + \sin \alpha_n \cdot \sin 5^\circ = \cos(\alpha_n - 5^\circ)$$
$$\sin \alpha_{n+1} = \sin \alpha_n \cdot \cos 5^\circ - \cos \alpha_n \cdot \sin 5^\circ = \sin(\alpha_n - 5^\circ)$$

В итоге мы получили, что $\alpha_{n+1} = \alpha_n - 5^\circ$, поэтому $\alpha_{2016} = \alpha_0 - 2016 \cdot 5^\circ = \alpha_0$ ($2016 \cdot 5^\circ = 28 \cdot 360^\circ$). Следовательно, наш вектор совершит 28 полных оборотов по часовой стрелке и вернется в исходное положение

$$x_{2016} + y_{2016} = (1 - \sqrt{2}) + (1 + \sqrt{2}) = 2$$

Ответ: 2

1.6 Вторая попытка
Задачи по информатике

Задача 1.6.1 (1 балл)

Условие:

Вы попали на завод роботов! Все роботы на заводе пронумерованы натуральными числами, начиная с 1. Так как производство не останавливается ни на секунду, то можно считать, что количество роботов бесконечно. Назовем пару роботов комбинируемой, если сумма номеров роботов не превышает k.

Вам дано число k, требуется посчитать количество неупорядоченных комбинируемых пар роботов. Пара называется неупорядоченной, если для ее обозначения не важен порядок (т.е пары (1, 2) и (2, 1) не различаются).

Формат входных данных:

Задано единственное число k $(1 \leq k \leq 30\,000)$.

Формат выходных данных:

Выведите единственное число - количество неупорядоченных комбинируемых пар роботов.

Пример:

stdin:
5
stdout:
4

Пояснение:

В тесте из примера возможны следующие пары: (1, 2), (1, 3), (1, 4), (2, 3).

Решение:

Посмотрим сначала количество пар роботов, чтобы их сумма была ровно k, таких чисел $\frac{k-1}{2}$, таким образом нам нужно посчитать $\sum_{1}^{k} \frac{k-1}{2}$, что является арифметической прогрессией.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    k = int(dataset)
    return str((k - 1) // 2 * (k // 2))

print(solve(sys.stdin.read()))
```

Задача 1.6.2 (2 балла)

Условие:

Для управления роботами центральный компьютер посылает им команды, состоящие из символом латинского алфавита. Количество операций растет каждый день и они уже не вмещаются в память роботов. Поэтому вам поручили реализовать метод сжатия строк, основанный на повторяющихся символах.

Сжатие строки происходит по следующему алгоритму: изначальная строка делится на отрезки максимальной длины, состоящие из одинаковых символов. После чего каждый отрезок заменяется на сам повторяющийся символ и количество его повторов, но при этом, если символ повторяется только один раз, то количество его повторов не выписывается.

Формат входных данных:

Единственная строка входных данных содержит саму строку, состоящую только из строчных символов латинского алфавита, длина строки не превосходит 50000.

Формат выходных данных:

Выведите сжатую строку.

Пример №1:

stdin:
aabccc
stdout:

a2bc3

Пример №2:

stdin:
abcd
stdout:
abcd

Решение:

Запомним указатель на текущий элемент, после чего будем сдвигать его, пока символы одинаковые. Осталось посчитать длину отрезка и вывести в нужном формате. Продолжаем, пока не дойдем до конца строки. Не забываем, что в некоторых языках строки являются неизменяемыми, поэтому при добавление нового символа строка будет пересоздаваться заново.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    s = dataset.split()[0]
    res = []
    i = 0
    while i < len(s):
        j = i
        while j < len(s) and s[i] == s[j]:
            j += 1
        res.append(s[i])
        if not j - i == 1:
            res.append(j - i)
        i = j
    return ''.join(str(x) for x in res)

print(solve(sys.stdin.read()))
```

Задача 1.6.3 (2 балла)

Условие:

В логистическом центре, где всю работу выполняют роботы – очень длинные стеллажи. На стеллаже умещается n коробок с грузами. Робот-погрузчик может манипулировать одновременно только с 1 или 2 коробками. За какое минимальное количество манипуляций робот сможет разгрузить весь стеллаж от грузов, при условии, что данное количество манипуляций должно быть кратно числу m из-за особенностей системы управления роботами-погрузчиками.

Формат входных данных:

Единственная строка входных данных содержит числа n и m $(1 \leq n \leq 50\,000, 1 \leq m \leq 100)$.

Формат выходных данных:

Выведите одно число — минимальное количество манипуляций, кратное m. Если такого числа не существует, тогда выведите -1.

Пример:

stdin:
10 2
stdout:
6

Решение:

Если мы знаем количество операций с двумя коробками, то $n - 2 \cdot i$ операций будут с одной коробкой. Поэтому переберем количество двойных манипуляций, и вычислив количество одинарных запомним минимальный ответ.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    mx = 0xffffffffff
    n, m = [int(x) for x in dataset.split()]
    ans = mx
    for i in range(0, n // 2 + 1):
        if (i + (n - 2 * i)) % m == 0:
            ans = min(ans, i + (n - 2 * i))
    return "-1" if ans == mx else str(ans)

print(solve(sys.stdin.read()))
```

Задача 1.6.4 (2 балла)

Условие:

В новом почтовом отделении вместо доставки писем по адресу, решили доставлять их в абонентские ящики, пронумерованные от 1 до n. У почтового робота есть операция доставки рассылки в абонентские ящики с номерами от L до R, включительно. Он добавляет ровно один буклет в каждый ящик. Известно, что операцию доставки использовали k раз. Помогите руководству узнать максимальное количество буклетов в ящиках.

Формат входных данных:

Первая строка входных данных содержит два числа n и k - количество абонентских ящиков и рассылок буклетов $(1 \leq n \leq 1\,000\,000, 1 \leq k \leq 25\,000)$. Следующие k строк содержат по два числа L и R, границы операции для доставки рассылок $(1 \leq L \leq R \leq n)$.

Формат выходных данных:

Выведите единственное число - максимальное количество буклетов в ящике.

Пример:

stdin:
7 4
5 5
2 4
4 6
3 5
stdout:
3

Пояснение:

Абонентские ящики из примера: [0, 1, 2, 3, 3, 1, 0].

Решение:

Решение “в лоб” за O(n · k) операций не подходит по времени, поэтому такое решение не рассматриваем. Заметим, что одну операцию мы можем разбить на две другие: добавить 1 ко всем элементам, начиная с L, и вычесть 1 у всех элементов начиная с R + 1. Таким образом мы можем посчитать разность между всеми соседними элементами, более того, зная такие разности мы можем восстановить наш массив. После чего остается найти максимум в таком массиве.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

def solve(dataset):
    arr = [int(x) for x in dataset.split()]
    n, k, arr = arr[0], arr[1], arr[2:]
    a = [0] * (n + 2);
    for i in range(0, k):
        l, r = arr[2 * i] - 1, arr[2 * i + 1] - 1
        a[l] += 1
        a[r + 1] -= 1
    val = 0
    ans = 0
    for i in range(0, n):
        val += a[i]
        a[i] = val;
        ans = max(ans, val)
    return str(ans)

print(solve(sys.stdin.read()))
```

Задача 1.6.5 (3 балла)

Условие:

Логистический центр представляет из себя сложную систему переходов между различными помещениями, по которым перемещаются роботы-погрузчики. Все помещения пронумерованы от 1 до n. Один из роботов получил задачу перенести груз из первого помещения в последнее. Время перемещения от входа в одно помещение до входа в другое занимает 1 минуту, внутри помещения робот перемещается мгновенно. Для перемещения по маршруту роботу не обязательно посещать все помещения, какие есть в центре, но он должен обязательно посетить помещение с номером k, где происходит контроль целостности груза, проверка также происходит мгновенно.

Ваша задача найти минимальное время, которое робот потратит на прохождение от первого помещения до последнего, заглянув на проверку целостности груза. Если пути не существует, выведите -1.

Формат входных данных:

Первая строка входных данных содержит три числа n, m, k - количество помещений, количество переходов, соединяющих помещения, и номер помещения проверки целостности груза, соответственно

$$(3 \leq n \leq 10^5, 2 \leq m \leq \min(n \cdot \frac{n-1}{2}, 10^5), 2 \leq k < n).$$

В следующих m строках заданы пары чисел v и u - номера помещений, которые соединены двусторонним переходом $(1 \leq v \leq u \leq n)$. Никакие два помещения не соединены двумя переходами, и переход не соединяет помещение с самим с собой.

Формат выходных данных:

Выведите единственное число - минимальное время, необходимое для прохождения маршрута, или -1, если это невозможно.

Пример:

stdin:

5 5 3
1 2
1 3
2 3
3 4
4 5

stdout:

3

Решение:

Найдем сначала кратчайшее расстояние от первого помещения до помещения с номером k, сделать это можно алгоритмом поиска в ширину. Таким же образом найдем кратчайшее расстояние от помещения k до помещения n.

Пример программы, реализующей данный алгоритм на языке Python:

```
import sys

INF = 10 ** 6

def bfs(n, start, finish, edges):
    d = [INF for i in range(n)]
    d[start] = 0
    q = deque()
    q.append(start)
    while len(q) > 0:
        u = q.popleft()
        for v in edges[u]:
            if d[u] + 1 < d[v]:
                d[v] = d[u] + 1
                q.append(v)
    return d[finish]

def solve(dataset):
    s = dataset.splitlines()
    n, m, k = map(int, s[0].split())
    edges = [[] for i in range(n)]

    for i in range(1, m + 1):
        u, v = map(int, s[i].split())
        edges[u - 1].append(v - 1)
        edges[v - 1].append(u - 1)

    res1 = bfs(n, 0, k - 1, edges)
    res2 = bfs(n, k - 1, n - 1, edges)
    if res1 == INF or res2 == INF:
```

```
        return "-1\n"
    return str(res1 + res2) + "\n"

print(solve(sys.stdin.read()))
```