1. ПЕРВЫЙ ЭТАП

Задачи первого этапа. Математика.

1.1. Первая попытка. Задачи 9 класса.

Задача 1.1.1. (20 баллов)

Гоша взял у друга 11 гаек М6 (ГОСТ 5916-70) и положил в карман рюкзака. Согласно ГОСТу 1 гайка М6 весит 1.254 грамма. И вот незадача, придя домой, Гоша насчитал в кармане 12 внешне одинаковых гаек! Одна из них была из того набора, что когда-то был куплен на блошином рынке, и, по его личному опыту, такие гайки имеют меньший вес, около грамма, а также сами по себе более низкого качества менее прочные.

У Гошиного папы есть весы, состоящие из двух больших чаш на двух концах рычага. За какое минимальное количество взвешиваний можно найти ту самую низкокачесвенную гайку?

Решение

Если мы имеем n внешне одинаковых объектов, после одного взвешивания останется в худшем случае $\lceil n \rceil$ объектов, среди которых есть отличающийся по весу. Таким образом, если $n = 12$, то, в худшем случае, понадобятся 3 взвешивания $12 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1$.

Ответ: 3.

Задача 1.1.2. (20 баллов)

Витя приклеивал цифры номера квартиры на дверь. Этот номер состоит из трех цифр. В процессе приклеивания Вите пришла необычная мысль, что если в номере квартиры поменять местами две последние цифры и сложить получившееся число с исходным, то получится номер его школы! Юноша учился в школе 1187. Найдите все такие номера квартир, и если Витя живет в квартире с наименьшим из них, то в какой квартире он живет?

Решение

Пусть номер квартиры равен $\overline{abc} = 100 \cdot a + 10 \cdot b + c$, где a, b и c – цифры числа. Число с перевернутыми двумя последними цифрами при этом будет равно

$\overline{acb} = 100 \cdot a + 10 \cdot c + b$. Затем решим уравнение в целых числах

$$100 \cdot a + 10 \cdot b + c + 100 \cdot a + 10 \cdot c + b = 1187,$$

$$200 \cdot a + 11 \cdot b + 11 \cdot c = 1187$$

с учетом ограничений на то, что a, b и c – цифры в десятичной системе счисления.

$$200 \cdot a + 11 \cdot (b + c) = 200 \cdot 5 + 11 \cdot 17.$$

Таким образом, возможные номера квартир 589 и 598. Выбираем наименьший – 589.

Ответ: 589.

Задача 1.1.3. (20 баллов)

Маша и Андрей, будущие математики, развлекались на перемене. Маша написала на доске 4 различных натуральных числа. Андрей выписал значения наибольших общих делителей для каждой из шести пар чисел. Получилось, что для одной из пар НОД равен 1, для другой — 2, для третьей — 3, для четвертой — 4, для пятой — 5, а для шестой — X. Найдите наименьшее возможное значение X?

Решение

Пусть на доске записаны 4 числа a, b, c и d. Пусть НОД$(a, b) = 2$, тогда НОД$(c, d) = 4$ быть не может, так как НОД всех пар будет кратным 2. Значит, пусть НОД$(a, c) = 4$ и тогда d будет нечётным числом.

Исходя из записанных равенств, можно выписать следующие:

$$a = 4 \cdot a_4 = 2 \cdot a_2$$

$$b = 2 \cdot b_2$$

$$c = 4 \cdot c_4$$

При этом НОД$(a_2, b_2) = 1$ и НОД$(a_4, c_4) = 1$. Очевидно также, что НОД$(b, c) = 2 \cdot x$ – то есть это последняя искомая пара. Попробуем подобрать значения a, b, c и d так, чтобы они удовлетворяли всем равенствам и из НОД оставшихся пар равнялись указанным значениям. Например, $a = 4$, $b = 10$, $c = 12$ и $d = 15$. Таким образом, наименьшее возможное значение равно 2.

Ответ: 2.

Задача 1.1.4. (20 баллов)

Женя загадал некоторое число: его наименьший делитель (не равный 1) на 77 меньше наибольшего делителя (не равного самому числу). Чему равно это число? Укажите наименьшее из возможных.

Решение

Пусть загаданное число $X = A \cdot B$, где A – наименьший делитель, B – наибольший. Тогда $A{+}77 = B$. $X = A{\cdot}(A{+}77) \to min$. Так как $A \neg 1$, то минимум достигается при $A = 2$. Таким образом, $X = 2 \cdot 79 = 158$.

Ответ: 158.

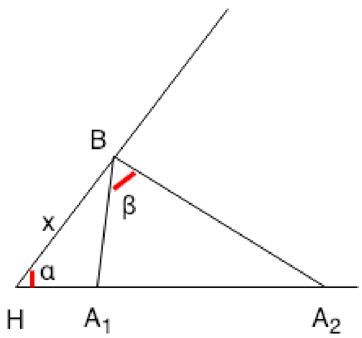
Задача 1.1.5. (20 баллов)

Дан некоторый острый угол $\alpha = 60°$. На одной из его сторон отмечены точки A_1 и A_2, на другой стороне отмечена точка B.

Вершина угла — H. Известно, что $HA_1 = 2$, $A_1A_2 = 8$. При какой величине отрезка HB величина острого угла между прямыми A_1B и A_2B будет максимальна? Ответ введите с точностью до десятитысячных.

Решение

Обозначим $\alpha = \angle BAA_2 = 60°$, $\beta = \angle A_1BA_2$, $x = BH$.



По теореме косинусов выразим A_1B и A_2B и подставим значения $A_1 = 2B$ и $A_1A_2 = 8$ из условия:

$$A_1B^2 = x^2 + A_1H^2 - 2xA_1H \cdot \cos\alpha = x^2 + 4 - 2x,$$
$$A_2B^2 = x^2 + A_2H^2 - 2xA_2H \cdot \cos\alpha = x^2 + 100 - 10x.$$

Рассмотрим $\triangle A_1BA_2$, по теореме косинусов:

$$A_1A_2{}^2 = A_1B^2 + A_2B^2 - 2A_1B \cdot A_2B\cos\beta.$$

Выразим $\cos\beta$:

$$\cos\beta = \frac{A_1B^2 + A_2B^2 - A_1A_2{}^2}{2A_1B \cdot A_2B} = \frac{x^2 + 4 - 2x + x^2 + 100 - 10x - 64}{2\sqrt{x^2+4-2x} \cdot \sqrt{x^2+100-10x}} =$$
$$= \frac{x^2 - 6x + 20}{\sqrt{x^2+4-2x} \cdot \sqrt{x^2+100-10x}}.$$

Для максимизации острого угла A_1BA_2 требуется, найти x, при котором достигается $min(\cos\beta)$. Решим уравнение $(\cos\beta)'_x = 0$ и найдем точку минимума. $x = 2\sqrt{5} \approx 4.472135955$.

Ответ: 4.472135955.

1.2. Первая попытка. Задачи 10-11 класса.

Задача 1.2.1. (10 баллов)

Гоша взял у друга 11 гаек M6 (ГОСТ 5916-70) и положил в карман рюкзака. Согласно ГОСТу 1 гайка M6 весит 1.254 грамма. И вот незадача, придя домой, Гоша насчитал в кармане 12 внешне одинаковых гаек! Одна из них была из того набора, что когда-то был куплен на блошином рынке, и, по его личному опыту, такие гайки имеют меньший вес, около грамма, а также сами по себе более низкого качества, менее прочные.

У Гошиного папы есть весы, состоящие из двух больших чаш на двух концах рычага. За какое минимальное количество взвешиваний можно найти ту самую низкокачесвенную гайку?

Решение

Если мы имеем n внешне одинаковых объектов, после одного взвешивания останется в худшем случае $\lceil n \rceil$ объектов, среди которых есть отличающийся по весу. Таким образом, если $n = 12$, то, в худшем случае, понадобятся 3 взвешивания $12 \Rightarrow 4 \Rightarrow 2 \Rightarrow 1$.

Ответ: 3.

Задача 1.2.2. (10 баллов)

Витя приклеивал цифры номера квартиры на дверь. Этот номер состоит из трех цифр. В процессе приклеивания Вите пришла необычная мысль, что если в номере квартиры поменять местами две последние цифры и сложить получившееся число с исходным, то получится номер его школы! Юноша учился в школе 1187. Найдите все такие номера квартир, и если Витя живет в квартире с наименьшим из них, то в какой квартире он живет?

Решение

Пусть номер квартиры равен $\overline{abc} = 100 \cdot a + 10 \cdot b + c$, где a, b и c – цифры числа. Число с перевернутыми двумя последними цифрами при этом будет равно $\overline{acb} = 100 \cdot a + 10 \cdot c + b$. Затем решим уравнение в целых числах

$$100 \cdot a + 10 \cdot b + c + 100 \cdot a + 10 \cdot c + b = 1187,$$

$$200 \cdot a + 11 \cdot b + 11 \cdot c = 1187$$

с учетом ограничений на то, что a, b и c – цифры в десятичной системе счисления.

$$200 \cdot a + 11 \cdot (b + c) = 200 \cdot 5 + 11 \cdot 17.$$

Таким образом, возможные номера квартир 589 и 598. Выбираем наименьший – 589.

Ответ: 589.

Задача 1.2.3. (10 баллов)

В химико-биологическом классе 25 учащихся. Для дежурства по школе всегда наугад выбирают двоих. Вероятность того, что оба дежурных окажутся мальчиками, равна 3/25. Сколько в классе девочек?

Решение

Пусть N – количество учащихся в классе, A – количество юношей в классе. Вероятность того, что оба дежурных окажутся мальчиками составляет

$$P = \frac{A}{N} \cdot \frac{A-1}{N-1}.$$

Подставим из условия $N = 25$, $P = 3/25$ и получим следующее уравнение:

$$\frac{A}{25} \cdot \frac{A-1}{24} = \frac{3}{25},$$

$$A^2 - A - 3 \cdot 24 = 0.$$

Уравнение имеет только один положительный корень $A = 9$. Значит, девочек в классе $25 - 9 = 16$.

Ответ: 16.

Задача 1.2.4. (10 баллов)

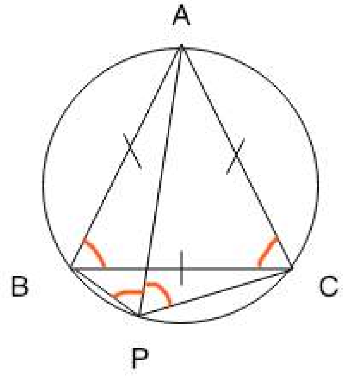
На уроке геометрии нарисовали окружность. На дуге BC этой окружности, описанной около равностороннего треугольника ABC, взята произвольная точка P. Выразите отрезок AP через отрезки и .

Укажите длину AP, если $BP = 3$, $CP = 4$. Ответ введите с точностью до десятитысячных.

Решение

Так как $\triangle ABC$ – равносторонний, то $\angle BAC = \angle ABC = \angle ACB = 60°$ и $AB = AC$. $\angle ABC = \angle APC$, так как опираются на $\smile AC$. $\angle ACB = \angle APB$, так как опираются на $\smile AB$. Таким образом, $\angle APB = \angle APC = 60°$. По теореме косинусов

$$\begin{cases} AB^2 = AP^2 + BP^2 - 2 \cdot AP \cdot BP \cdot \cos(\angle APB), \\ AC^2 = AP^2 + CP^2 - 2 \cdot AP \cdot CP \cdot \cos(\angle APC). \end{cases}$$



Так как $AB = AC$ и $\angle APC = \angle APB = 60^\circ$, получим:

$$AP^2 + BP^2 - AP \cdot BP \cdot = AP^2 + CP^2 - AP \cdot CP,$$
$$BP^2 - CP^2 = AP \cdot (BP - CP),$$
$$AP = BP - CP.$$

Подставим значения $BP = 3$ и $CP = 4$ и получим $AP = 7$.

Ответ: 7.

Задача 1.2.5. (10 баллов)

Маша и Андрей, будущие математики, развлекались на перемене. Маша написала на доске 4 различных натуральных числа. Андрей выписал значения наибольших общих делителей для каждой из шести пар чисел. Получилось, что для одной из пар НОД равен 1, для другой — 2, для третьей — 3, для четвертой — 4, для пятой — 5, а для шестой — X. Найдите наименьшее возможное значение X?

Решение

Пусть на доске записаны 4 числа a, b, c и d. Пусть НОД$(a, b) = 2$, тогда НОД$(c, d) = 4$ быть не может, так как НОД всех пар будет кратным 2. Значит, пусть НОД$(a, c) = 4$ и тогда d будет нечётным числом.

Исходя из записанных равенств, можно выписать следующие:

$$a = 4 \cdot a_4 = 2 \cdot a_2$$

$$b = 2 \cdot b_2$$

$$c = 4 \cdot c_4$$

При этом НОД$(a_2, b_2) = 1$ и НОД $(a_4, c_4) = 1$.

Очевидно также, что НОД$(b, c) = 2 \cdot x$ – то есть это последняя искомая пара. Попробуем подобрать значения a, b, c и d так, чтобы они удовлетворяли всем равенствам и из НОД оставшихся пар равнялись указанным значениям. Например, $a = 4$, $b = 10$, $c = 12$ и $d = 15$. Таким образом, наименьшее возможное значение равно 2.

Ответ: 2.

Задача 1.2.6. (10 баллов)

Женя загадал некоторое число: его наименьший делитель (не равный 1) на 77 меньше наибольшего делителя (не равного самому числу). Чему равно это число? Укажите наименьшее из возможных.

Решение

Пусть загаданное число $X = A \cdot B$, где A – наименьший делитель, B – наибольший. Тогда $A+77 = B$. $X = A \cdot (A+77) \to min$. Так как $A \neg 1$, то минимум достигается при $A = 2$. Таким образом, $X = 2 \cdot 79 = 158$.

Ответ: 158.

Задача 1.2.7. (10 баллов)

У Лады на прикроватной тумбочке стоят часы с циферблатом. Они показывают текущее время суток от 00.00.00 до 23.59.59. Однако, сосед Дима решил перепрошить часы, и теперь, если на часах должны загореться ровно четыре цифры 3, циферблат перестает гореть. Сколько времени в течение суток часы не показывают время, если всё остальное время они работают корректно? Ответ укажите в секундах.

Решение

Каждая комбинация цифр на циферблате отображается ровно 1 секунду, следовательно в задаче требуется найти количество соответствующих комбинаций. В группе цифр, отображающей часы может встретиться только 0 или 1 тройка. В остальных – 0, 1 или 2. Четыре тройки можно получить из следующих комбинаций: $X_1 = Q([0] : [2] : [2])$, $X_2 = Q([1] : [1] : [2])$, $X_3 = Q([1] : [2] : [1])$. Q – количество комбинаций в соответствии с количеством троек в группах цифр циферблата, соответствующих часам, минутам, секундам. $X_1 = 21 \cdot 1 \cdot 1$. $X_2 = 3 \cdot 14 \cdot 1$. $X_3 = 3 \cdot 1 \cdot 14$. $X = X_1 + X_2 + X_3 = 21 + 42 + 42 = 105$.

Ответ: 105.

Задача 1.2.8. (10 баллов)

Дано уравнение $x^3 + y^3 = 4(x^2y + xy^2 + 1)$. Сколько различных решений в целых числах оно имеет?

Если решений бесконечное множество, введите -1.

Решение

$$x^3 + y^3 = 4(x^2y + xy^2 + 1),$$
$$(x + y)(x^2 - xy + y^2) - 4xy(x + y) = 4,$$

$$(x+y)(x^2-5xy+y^2)=4.$$

В левой части множители могут принимать только следующие пары значений: (-4, 1), (-2, -2), (-1, -4), (1, 4), (2, 2), (4, 1). Все 6 систем уравнений не дают целых корней.

Ответ: 0.

Задача 1.2.9. (10 баллов)

Дан некоторый острый угол $\alpha = 60°$. На одной из его сторон отмечены точки A_1 и A_2, на другой стороне отмечена точка B.

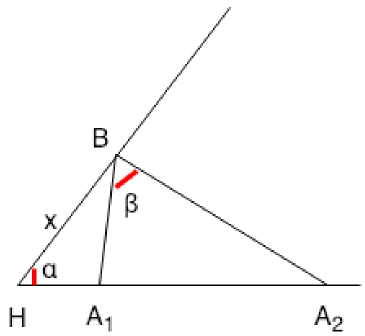
Вершина угла — H. Известно, что $HA_1 = 2$, $A_1A_2 = 8$.

При какой величине отрезка HB величина острого угла между прямыми A_1B и A_2B будет максимальна?

Ответ введите с точностью до десятитысячных.

Решение

Обозначим $\alpha = \angle BAA_2 = 60°$, $\beta = \angle A_1BA_2$, $x = BH$.



По теореме косинусов выразим A_1B и A_2B и подставим значения $A_1 = 2B$ и $A_1A_2 = 8$ из условия:

$$A_1B^2 = x^2 + A_1H^2 - 2xA_1H \cdot \cos\alpha = x^2 + 4 - 2x,$$

$$A_2B^2 = x^2 + A_2H^2 - 2xA_2H \cdot \cos\alpha = x^2 + 100 - 10x.$$

Рассмотрим $\triangle A_1BA_2$, по теореме косинусов:

$$A_1A_2{}^2 = A_1B^2 + A_2B^2 - 2A_1B \cdot A_2B\cos\beta.$$

Выразим $\cos\beta$:

$$\cos\beta = \frac{A_1B^2 + A_2B^2 - A_1A_2{}^2}{2A_1B \cdot A_2B} = \frac{x^2+4-2x+x^2+100-10x-64}{2\sqrt{x^2+4-2x}\cdot\sqrt{x^2+100-10x}} =$$

$$= \frac{x^2-6x+20}{\sqrt{x^2+4-2x}\cdot\sqrt{x^2+100-10x}}.$$

Для максимизации острого угла A_1BA_2 требуется, найти x, при котором достигается $min(\cos\beta)$. Решим уравнение $(\cos\beta)'_x = 0$ и найдем точку минимума. $x = 2\sqrt{5} \approx 4.472135955$.

Ответ: 4.472135955.

Задача 1.2.10. (10 баллов)

Коптер летит над поверхностью огромного поля.

В какой-то момент времени он оказывается в точке $(0, 3, 6)$ в заданной ортогональной системе координат с осями Ox, Oy и Oz.

Найдите расстояние от коптера до земли, если в той же системе координат поле можно считать плоскостью, заданной уравнением $2x+4y-4z-6 = 0$. Ответ укажите с точностью до десятитысячных.

Решение

Расстояние между точкой с координатами (x_0, y_0, z_0) и плоскостью, задаваемой уравнением $Ax + By + Cx + D = 0$ вычисляется по формуле:

$$S = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}}$$

Для заданных точки и плоскости

$$S = \frac{|2 \cdot 0 + 4 \cdot 3 - 4 \cdot 6 - 6|}{\sqrt{2^2 + 4^2 + 4^2}} = 3.$$

Ответ: 3.

1.3. Вторая попытка. Задачи 9 класса.

Задача 1.3.1. (20 баллов)

Полина прячет в кулаке от 1 до 4 гвоздей. Валера пытается угадать, сколько их. Для этого он задаёт вопросы, на которые Полина может отвечать "да"и "нет". За какое минимальное число вопросов Валера может угадать количество спрятанных гвоздей.

Решение

За один вопрос можно сократить перебираемое множество вдвое. Таким образом, достаточно двух вопросов.

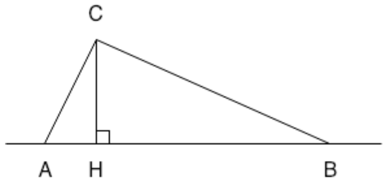
Ответ: 2.

Задача 1.3.2. (20 баллов)

В стране X есть три города – A, B, C. Известно, что расстояние между $AB = 25$, $CB = 24$, $AC = 7$. Города A и B лежат на прямолинейной границе страны, а все остальные участки границы страны пренебрежительно далеки. Найдите кратчайший путь из города до границы страны X. Ответ укажите с точностью до десятитысячных.

Решение

Наикратчайший путь равен длине перпендикуляра, опущенного из вершины C к прямой AB.



По длинам сторон очевидно, что $\triangle ABC$ – прямойгольный, с прямым углом в вершине $C(7^2 + 24^2 = 25^2)$. Так, согласно свойству прямоугольного треугольника, $CH = AC \cdot BC/AB = 6.72$.

Ответ: 6.72.

Задача 1.3.3. (20 баллов)

Коля – очень любознательный юноша. Он решил провести исследование. Для различных действительных чисел a он решил найти такое наибольшее целое число x, чтобы выполнялось следующее:

1. a лежит в интервале $(1, 2)$,
2. a^2 лежит в $(2, 3)$,
3. a^3 лежит в $(3, 4)$,
4. и так далее до показателя степени x.

Помогите Коле выяснить, каким же может быть максимальное значение числа x, при котором существует хотя бы одно значение a, удовлетворяющее условиям?

Решение

В данной задаче необходимо найти такое натуральное максимальное n, при котором $\exists a \forall x \in N : x < n \Rightarrow x < a^x < x + 1$. Выпишем границы интервалов $(\sqrt[x]{x}, \sqrt[x]{x+1})$ для нескольких первых значений x.

x	Левая граница a	Правая граница a
1	1	2
2	$\sqrt[2]{2} = 1.41421356$	$\sqrt[2]{3} = 1.73205081$
3	$\sqrt[3]{3} = 1.44224957$	$\sqrt[3]{4} = 1.58740105$
4	$\sqrt[4]{4} = 1.41421356$	$\sqrt[4]{5} = 1.49534878$
5	$\sqrt[5]{5} = 1.37972966$	$\sqrt[5]{6} = 1.43096908$
...	...	...

По таблице видно, что $\sqrt[5]{6} < \sqrt[3]{3}$. Таким образом, $n = 4$ – наибольшее значение, удовлетворяющее условию задачи.

Ответ: 4.

Задача 1.3.4. (20 баллов)

Найдите сумму коэффициентов после раскрытия скобок у выражения

$$(x^2 - 3x + 1)^{100}.$$

Решение

Постепенно раскрывая скобки найдем сумму коэффициентов полинома $(x^2 - 3x + 1)^n$ для n, начиная с 1.

n	Сумма коэффициентов
1	-1
2	1
3	-1
4	1
...	...

Подметим, что при нечётном n сумма коэффициентов равна -1, а при чётном – 1.

Ответ: 1.

Задача 1.3.5. (20 баллов)

На доске записаны 5 чисел: сначала некоторое рациональное $a = n/y$ (n и y натуральные взаимно простые числа), затем x и далее $x+2$, $x+3$ и $x+4$. При каком наименьшем значении a произведение всех пяти чисел всегда будет натуральным для любого натурального x? В ответе напишите целое число y.

Решение

В данной задаче надо проверить делимость произведения чисел. $x \cdot (x+2) \cdot (x+3) \cdot \cdot (x+4)$ гарантированно делится на 2 и на 3, так как содержит произведение последовательно идущих 3 чисел. Делимость на остальные числа при любом x гарантировать нельзя.

Ответ: 6.

1.4. Вторая попытка. Задачи 10-11 класса.

Задача 1.4.1. (10 баллов)

Полина прячет в кулаке от 1 до 4 гвоздей. Валера пытается угадать, сколько их. Для этого он задаёт вопросы, на которые Полина может отвечать "да"и "нет". За какое минимальное число вопросов Валера может угадать количество спрятанных гвоздей.

Решение

За один вопрос можно сократить перебираемое множество вдвое. Таким образом, достаточно двух вопросов.

Ответ: 2.

Задача 1.4.2. (10 баллов)

Мотоциклист поднимается на холм. Его движение в ортогональной системе координат xOy можно описать законом $y = ax^2+bx+c$, где a и b – некоторые неизвестные постоянные коэффициенты. Известно, что во время своего движения мотоциклист побывал в точках с координатами $(0, 1)$, $(1, 2)$, $(2, 1)$. Найдите координаты вершины холма. Ответ укажите в формате "(x, y)" где x и — значения абсциссы и ординаты с точностью до десятитысячных.

Решение

Подставим точки в уравнение и решим систему:

$$\begin{cases} 1 = a \cdot 0^2 + b \cdot 0 + c, \\ 2 = a \cdot 1^2 + b \cdot 1 + c, \\ 1 = a \cdot 2^2 + b \cdot 2 + c; \end{cases}$$

$$\begin{cases} c = 1, \\ a + b = 1, \\ 2a + b = 0; \end{cases}$$

$$\begin{cases} a = -1, \\ b = 2, \\ c = 1. \end{cases}$$

Уравнение параболы с вычисленными коэффициентами $y = -x^2+2x+1$. Найдем абсциссу точки перегиба:

$$y'_x = 0 \Leftrightarrow -2x + 2 = 0 \Leftrightarrow x = 1.$$

При x = 1 y = 2.

Ответ: $(1, 2)$.

Задача 1.4.3. (10 баллов)

Для тестирования новой программы компьютер выбирает случайное действительное число A из отрезка $[1, 2]$ и заставляет программу решать уравнение $3x + A = 0$. Учтите, что распределение случайной величины равномерное. Найдите вероятность того, что корень этого уравнения меньше, чем -0.4. Ответ укажите с точностью до десятитысячных.

Решение

Выразим корень уравнения: $x = -A/3$. Таким образом, $x \in [-2/3, -1/3]$. Значит

$$P(x \in [-2/3, -0.4]) = \frac{-0.4 + 2/3}{1/3} = 2 - 1.2 = 0.8.$$

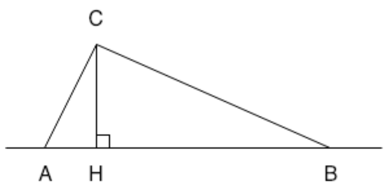
Ответ: 0.8.

Задача 1.4.4. (10 баллов)

В стране X есть три города – A, B, C. Известно, что расстояние между $AB = 25$, $CB = 24$, $AC = 7$. Города A и B лежат на прямолинейной границе страны, а все остальные участки границы страны пренебрежительно далеки. Найдите кратчайший путь из города до границы страны X. Ответ укажите с точностью до десятитысячных.

Решение

Наикратчайший путь равен длине перпендикуляра, опущенного из вершины C к прямой AB.



По длинам сторон очевидно, что $\triangle ABC$ – прямойгольный, с прямым углом в вершине $C(7^2 + 24^2 = 25^2)$. Так, согласно свойству прямоугольного треугольника, $CH = AC \cdot BC/AB = 6.72$.

Ответ: 6.72.

Задача 1.4.5. (10 баллов)

Коля – очень любознательный юноша. Он решил провести исследование. Для различных действительных чисел a он решил найти такое наибольшее целое число x, чтобы выполнялось следующее:

1. a лежит в интервале $(1, 2)$,
2. a^2 лежит в $(2, 3)$,
3. a^3 лежит в $(3, 4)$,
4. и так далее до показателя степени x.

Помогите Коле выяснить, каким же может быть максимальное значение числа x, при котором существует хотя бы одно значение a, удовлетворяющее условиям?

Решение

В данной задаче необходимо найти такое натуральное максимальное n, при котором $\exists a \forall x \in N : x < n \Rightarrow x < a^x < x + 1$. Выпишем границы интервалов $(\sqrt[x]{x}, \sqrt[x]{x+1})$ для нескольких первых значений x.

x	Левая граница a	Правая граница a
1	1	2
2	$\sqrt[2]{2} = 1.41421356$	$\sqrt[2]{3} = 1.73205081$
3	$\sqrt[3]{3} = 1.44224957$	$\sqrt[3]{4} = 1.58740105$
4	$\sqrt[4]{4} = 1.41421356$	$\sqrt[4]{5} = 1.49534878$
5	$\sqrt[5]{5} = 1.37972966$	$\sqrt[5]{6} = 1.43096908$
...	...	...

По таблице видно, что $\sqrt[5]{6} < \sqrt[3]{3}$. Таким образом, $n = 4$ – наибольшее значение, удовлетворяющее условию задачи.

Ответ: 4.

Задача 1.4.6. (10 баллов)

На доске записаны 5 чисел: сначала некоторое рациональное $a = n/y$ (n и y натуральные взаимно простые числа), затем x и далее $x+2$, $x+3$ и $x+4$. При каком наименьшем значении a произведение всех пяти чисел всегда будет натуральным для любого натурального x? В ответе напишите целое число y.

Решение

В данной задаче надо проверить делимость произведения чисел. $x \cdot (x+2) \cdot (x+3) \cdot$ $\cdot (x+4)$ гарантированно делится на 2 и на 3, так как содержит произведение последовательно идущих 3 чисел. Делимость на остальные числа при любом x гарантировать нельзя.

Ответ: 6.

Задача 1.4.7. (10 баллов)

Двое бросают монету: один бросил её 8 раз, другой – 9 раз. Чему равна вероятность того, что у второго монета упала орлом большее число раз, чем у первого?

Монета никогда не падает на ребро.

Ответ укажите с точностью до десятитысячных.

Решение

Первый может выбросить 9 различных наборов (последовательность не учитывается). Если в наборе x орлов, то количество возможных вариантов второго игрока, удовлетворяющих условию $(9 - x)$. Всего возможных способов выкинуть наборы у обоих игроков: $9 \cdot 10 = 90$.

$$P = \frac{\sum\limits_{x=0}^{8} 9 - x}{90} = 0.5.$$

Ответ: 0.5.

Задача 1.4.8. (10 баллов)

Найдите сумму коэффициентов после раскрытия скобок у выражения

$$(x^2 - 3x + 1)^{100}.$$

Решение

Постепенно раскрывая скобки найдем сумму коэффициентов полинома $(x^2 - 3x + 1)^n$ для n, начиная с 1.

n	Сумма коэффициентов
1	-1
2	1
3	-1
4	1
...	...

Подметим, что при нечётном n сумма коэффициентов равна -1, а при чётном – 1.

Ответ: 1.

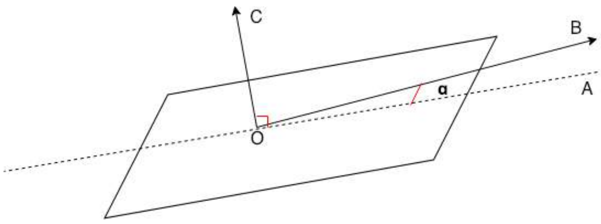
Задача 1.4.9. (10 баллов)

Самолет взлетает с авианосца. Вектор нормали к поверхности взлетной полосы имеет координаты $(4, 0, 3)$. Направляющий вектор траектории полета самолета – $(5, 12, 0)$ в той же ортогональной системе координат. Найдите СИНУС угла, под которым взлетел самолет относительно взлетной полосы.

Введите ответ с точностью до десятитысячных.

Решение

$$\cos\alpha = \frac{|(\vec{OB}, \vec{OC})|}{|\vec{OB}| \cdot |\vec{OC}|} = \frac{20}{\sqrt{25+144}\sqrt{16+9}} = \frac{4}{13},$$
$$\sin\alpha = \sqrt{1-\cos^2\alpha} = 0.9514859.$$



Ответ: 0.9514859136.

Задача 1.4.10. (10 баллов)

Соня, Андрей и Егор живут в домах A, B, C соответственно. Эти дома соединены прямыми улицами – Садовой, Огородной и Персиковой. Известно, что Садовая и Персиковая улицы пересекаются у дома Сони под углом 45 градусов, Персиковая и Огородная – под углом 60 градусов у дома Егора и, наконец, Огородная и Садовая пересекаются под окнами у Андрея. Равноудаленно от домов B и C внутри треугольника ABC построили магазин. Известно, что прямая улица, которая соединяет магазин и дом Егора, пересекается с Персиковой под углом 15 градусов. Между домом Сони и магазином также есть прямая улица. Под каким углом она пересекается с Садовой?

Ответ укажите в градусах с точностью до десятитысячных.

Решение

Обозначим искомый угол буквой α.

$\angle BAC = 45° \Rightarrow \angle OAC = 45° - \alpha$. $\angle OCA = 15° \Rightarrow \angle AOC = 120° + \alpha$. $\angle OCB = \angle OBC = 60° - 15° = 45° \Rightarrow \angle BOC = 90°$. $\angle ABO = 30°$. По теореме синусов:

$$\frac{\sin\alpha}{OB} = \frac{\sin 30°}{AO} \Rightarrow AO = \frac{OB}{2\sin\alpha}$$

$$\frac{\sin 15°}{OA} = \frac{\sin(45° - \alpha)}{OC} \Rightarrow AO = \frac{OC\sin 15°}{\sin(45° - \alpha)}$$

Так как $OB = OC$,

$$\sin(45° - \alpha) = 2\sin\alpha\sin 15° \Rightarrow \alpha = 30°.$$

Ответ: 30.

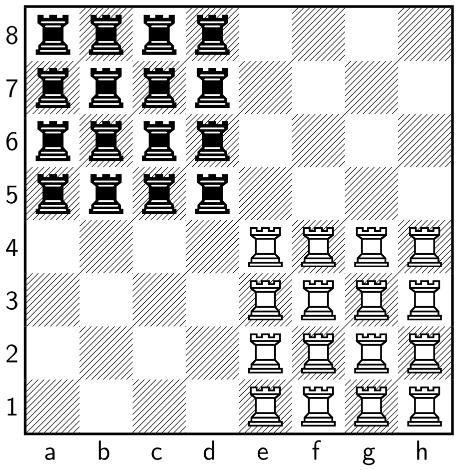
1.5. Третья попытка. Задачи 9 класса.

Задача 1.5.1. (20 баллов)

Пусть на шахматной доске размера 8×8 расставили n белых и столько же черных ладей так, что они не бьют друг друга. Найдите такое максимальное n, при котором это возможно.

Решение

Необходимо расставить ладьи одного цвета так, что на одной горизонтали с каждой из них было еще 3 и на одной вертикали 3 другие. Таким образом, в каждой строке или столбце ровно 4 ладьи одного цвета. Например:



Ответ: 16.

Задача 1.5.2. (20 баллов)

На планете Y все жители говорят на языке, состоящем из трех букв. При этом все слова в этом языке не длиннее 6 символов и не короче 3. Сколько слов можно составить в языке планеты Y?

Решение

Количество слов из 3 букв длиной n $W_n = 3^N$. Так $W = 3^3 + 3^4 + 3^5 + 3^6 = 1080$.

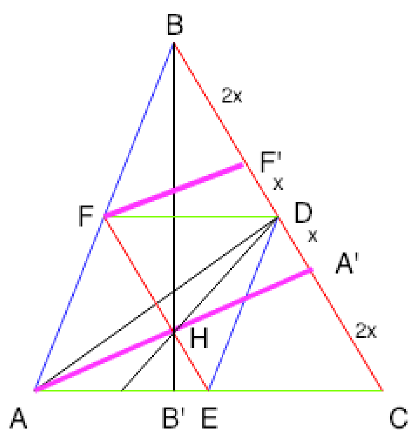
Ответ: 1080.

Задача 1.5.3. (20 баллов)

Дан треугольник ABC и H — точка пересечения высот этого треугольника. Пусть D — середина отрезка BC, E — середина отрезка . Кроме того, медианы треугольника AED пересекаются в точке H. Найдите градусную меру угла $\angle ABC$.

Ответ укажите с точностью до десятитысячных.

Решение



Рассмотрим $\triangle ABC$: DE, EF, DF – медианы, AA', BB' – высоты. $BD = DC$. В $\triangle ADE$: медианы пересекаются в точке H, которая разделяет их в отношении $2:1$. Из подобия треугольников очевидно, что $DA' : A'C = 1:2$.

Пусть $AD = x$, $A'C = 2x$. $FH = 2\cdot(HE + HE/2) - HE = 2HE = 2x$ По свойству паралелограмма, образованного медианами FE и BC, перпендикулярами AA' и FF': $FH = F'A' = 2x$ и $FF' = HA'$, $F'D = x$. Также из подобия треугольников $BF' = 2x$, $FF' = HA' = AH$.

Рассмотрим $\triangle BA'H \sim \triangle AHE$(по 2 углам). $A'B : A'H = A'H : HE \Rightarrow A'H = \sqrt{A'B\cdot HE} = \sqrt{4x^2} = 2x = EF' = BF'$.

Так, $\triangle BF'F$ – прямоугольный равнобедренный, и угол при основании $\angle FBF' = 45^\circ$.

Ответ: 45.

Задача 1.5.4. (20 баллов)

Артур и Саша играли в игру — по очереди выписывали натуральные числа на бумагу. В итоге оказалось, что на бумаге выписано 15 чисел, причем, наименьшее из чисел можно представить как $x+1$, $x>1$, а все остальные числа — последовательность $(1+x^n)$, где n — натуральный показатель степени, изменяющийся от 2 до 15. Артуру показалось, что выписанных чисел слишком много и он зачеркнул часть из них таким образом, чтобы все оставшиеся на бумаге числа были взаимно простыми. Какое наименьшее количество чисел мог зачеркнуть Артур?

Решение

Все натуральные числа вида $(1+x^n)$, где n – нечетное число, делятся нацело на $(1+x)$.

Таким образом, вычеркнем все числа, где $n = 1, 3, 5, 7, 9, 11$ и 13. 15 оставляем.

Далее заменим x^2 на y. По аналогии вычеркнем числа, где n = 2, 6, 10. 14 оставляем.

Далее заменим x^3 на z. $n = 3, 9$ – вычеркнуты. 15 оставляем.

Далее заменим x^4 на w. По аналогии вычеркнем число, где $n = 4$. 12 оставляем.

Далее заменим x^5 на v. $n = 5$ вычеркнуто. 15 оставляем.

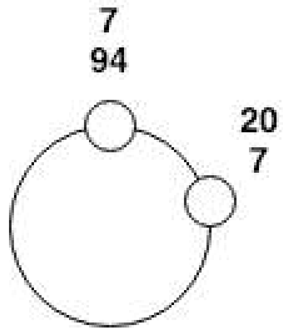
Далее нет смысла перебирать, так как $3n > 15$. Оставшиеся числа взаимнопростые. Таким образом, минимально мы вычеркнули 11 чисел (1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13).

Ответ: 11.

Задача 1.5.5. (20 баллов)

Ира и Паша расставляют стулья вокруг круглого стола. После того, как все стулья были расставлены, ребята решили их пересчитать — они начали ходить по кругу в одном направлении, но начиная с разных стульев. Известно, что стул, который Паша посчитал седьмым, у Иры оказался под двадцатым номером, а тот стул, который Ира посчитала седьмым, у Паши был 94 м. Сколько стульев было расставлено вокруг стола?

Решение



Считаем по обеим дугам количество стульев, включая один из двух крайних: $20 - 7 + 94 - 7 = 100$.

Ответ: 100.

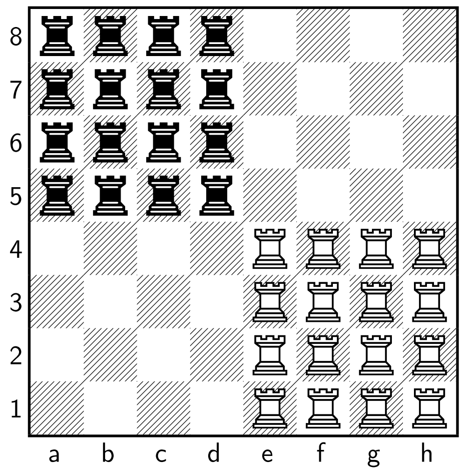
1.6. Третья попытка. Задачи 10-11 класса.

Задача 1.6.1. (10 баллов)

Пусть на шахматной доске размера 8×8 расставили n белых и столько же черных ладей так, что они не бьют друг друга. Найдите такое максимальное n, при котором это возможно.

Решение

Необходимо расставить ладьи одного цвета так, что на одной горизонтали с каждой из них было еще 3 и на одной вертикали 3 другие. Таким образом, в каждой строке или столбце ровно 4 ладьи одного цвета. Например:



Ответ: 16.

Задача 1.6.2. (10 баллов)

Найдите такое значение $a > 1$, при котором уравнение $a^x = log_a x$ имеет ровно один корень. Ответ укажите с точностью до десятитысячных.

Решение

Обе функции в левой и правой частях уравнения выпуклые, единственный корень у уравнения возникает тогда, когда $y = a^x$ касается прямой $y = x$, то есть $f(x_0) = x_0$ и $f'(x_0) = a^{x_0} \ln a = 1$. Отсюда $x_0 = \frac{1}{\ln a}$, т.е. $e = a^{\frac{1}{\ln a}} = \frac{1}{\ln a}$, $\ln a = \frac{1}{e}$, $a = e^{\frac{1}{e}} = 1.44466786$.

Ответ: 1.44466786101.

Задача 1.6.3. (10 баллов)

На планете Y все жители говорят на языке, состоящем из трех букв. При этом все слова в этом языке не длиннее 6 символов и не короче 3. Сколько слов можно составить в языке планеты Y?

Решение

Количество слов из 3 букв длиной n $W_n = 3^N$. Так $W = 3^3 + 3^4 + 3^5 + 3^6 = 1080$.

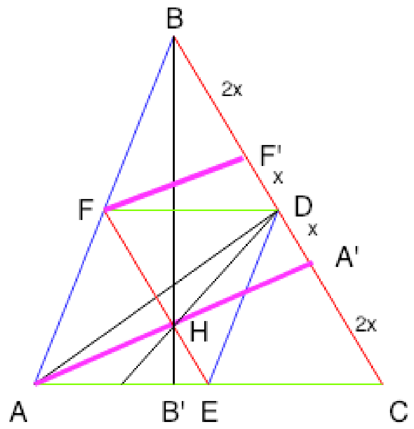
Ответ: 1080.

Задача 1.6.4. (10 баллов)

Дан треугольник ABC и H — точка пересечения высот этого треугольника. Пусть D — середина отрезка BC, E — середина отрезка . Кроме того, медианы треугольника AED пересекаются в точке H. Найдите градусную меру угла $\angle ABC$.

Ответ укажите с точностью до десятитысячных.

Решение



Рассмотрим $\triangle ABC$: DE, EF, DF – медианы, AA', BB' – высоты. $BD = DC$. В $\triangle ADE$: медианы пересекаются в точке H, которая разделяет их в отношении $2:1$. Из подобия треугольников очевидно, что $DA' : A'C = 1 : 2$.

Пусть $AD = x$, $A'C = 2x$. $FH = 2\cdot(HE + HE/2) - HE = 2HE = 2x$ По свойству паралелограмма, образованного медианами FE и BC, перпендикулярами AA' и FF': $FH = F'A' = 2x$ и $FF' = HA'$, $F'D = x$. Также из подобия треугольников $BF' = 2x$, $FF' = HA' = AH$.

Рассмотрим $\triangle BA'H \sim \triangle AHE$(по 2 углам). $A'B : A'H = A'H : HE \Rightarrow A'H = \sqrt{A'B \cdot HE} = \sqrt{4x^2} = 2x = EF' = BF'$.

Так, $\triangle BF'F$ – прямоугольный равнобедренный, и угол при основании $\angle FBF' = 45°$.

Ответ: 45.

Задача 1.6.5. (10 баллов)

Артур и Саша играли в игру — по очереди выписывали натуральные числа на бумагу. В итоге оказалось, что на бумаге выписано 15 чисел, причем, наименьшее из чисел можно представить как $x + 1$, $x > 1$, а все остальные числа — последовательность $(1 + x^n)$, где n — натуральный показатель степени, изменяющийся от 2 до 15. Артуру показалось, что выписанных чисел слишком много и он зачеркнул часть из них таким образом, чтобы все оставшиеся на бумаге числа были взаимно простыми. Какое наименьшее количество чисел мог зачеркнуть Артур?

Решение

Все натуральные числа вида $(1 + x^n)$, где n – нечетное число, делятся нацело на $(1 + x)$.

Таким образом, вычеркнем все числа, где n = 1, 3, 5, 7, 9, 11 и 13. 15 оставляем.

Далее заменим x^2 на y. По аналогии вычеркнем числа, где n = 2, 6, 10. 14 оставляем.

Далее заменим x^3 на z. $n = 3$, 9 – вычеркнуты. 15 оставляем.

Далее заменим x^4 на w. По аналогии вычеркнем число, где $n = 4$. 12 оставляем.

Далее заменим x^5 на v. $n = 5$ вычеркнуто. 15 оставляем.

Далее нет смысла перебирать, так как $3n > 15$. Оставшиеся числа взаимнопростые. Таким образом, минимально мы вычеркнули 11 чисел (1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13).

Ответ: 11.

Задача 1.6.6. (10 баллов)

Паша загадал число x: это неправильная дробь с натуральным числителем и со знаменателем, равным 9. Далее он вычислил еще три числа умножил x на 5, на 2 и на 4. Затем округлил эти три числа по правилам округления до целого и сложил между собой, в итоге он получил 120. Каким был числитель у неправильной дроби?

Решение

Найдем первое значение для перебора: $\frac{5x+2x+4x}{9} \approx 120 \Leftrightarrow x \approx 98$. Проверим значение:

$$\left[\frac{5 \cdot 98}{9}\right] + \left[\frac{2 \cdot 98}{9}\right] + \left[\frac{4 \cdot 98}{9}\right] = 54 + 22 + 44 = 120.$$

Ответ: 98.

Задача 1.6.7. (10 баллов)

Если из резервуара выливают воду, уровень воды H в нём меняется в зависимости от времени t следующим образом: $H(t) = at^2 + bt + c$. Пусть t_0 — момент окончания слива. Известно, что в этот момент выполнены равенства $H(t_0) = H'(t_0) = 0$. В течение какого времени вода из резервуара будет полностью вылита, если за первый час слилась половина уровня?

Округлите ответ до ближайшего целого.

Решение

$$H(t_0) = at_0^2 + bt_0 + c = 0,$$
$$H'(t_0) = 2at_0 + b = 0.$$

Выразим b и c через a и t_0:

$$b = -2at_0,$$
$$c = at_0^2.$$

Пусть τ – время достижения половинного уровня.

$$2H(\tau) = H(0),$$

$$2(a\tau^2 + (-2at_0)\tau + at_0^2) = at_0^2,$$
$$2\tau^2 - 4t_0\tau + t_0^2 = 0,$$
$$D = 16t_0^2 - 4 \cdot 2t_0^2 = 8t_0^2,$$
$$\tau = \frac{4t_0 \pm 2\sqrt{2}t_0}{2 \cdot 2} = t_0 \pm t_0/\sqrt{2}.$$

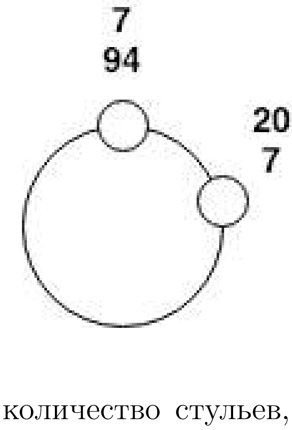
Так как $\tau < t_0$ из условия, то $\tau = t_0 \cdot (2 - \sqrt{2})/2$, то есть $t_0 = \tau(2 + \sqrt{2})$, то есть $[t_0] = [1 \cdot (2 + \sqrt{2})] = 3$.

Ответ: 3.

Задача 1.6.8. (10 баллов)

Ира и Паша расставляют стулья вокруг круглого стола. После того, как все стулья были расставлены, ребята решили их пересчитать — они начали ходить по кругу в одном направлении, но начиная с разных стульев. Известно, что стул, который Паша посчитал седьмым, у Иры оказался под двадцатым номером, а тот стул, который Ира посчитала седьмым, у Паши был 94 м. Сколько стульев было расставлено вокруг стола?

Решение



Считаем по обеим дугам количество стульев, включая один из двух крайних: $20 - 7 + 94 - 7 = 100$.

Ответ: 100.

Задача 1.6.9. (10 баллов)

В течение пяти часов 1 сентября Женя наблюдал за воздушными шариками в небе. По мере того, как утренние линейки проходили, шаров в небе становилось все меньше – так, с каждым часом за час пролетало не больше шаров, чем в предыдущий час. Суммарно Женя насчитал 100 шаров, пролетевших в небе мимо его окна. Причем, суммарно за второй и четвертый час Женя увидел не больше шаров, чем за первый и третий. Какое минимальное число шаров Женя мог увидеть суммарно за 1, 3 и 5 часы?

Решение

Пусть a, b, c, d, e — количество шаров в 1, 2, 3, 4 и 5 часы соответственно. $a \geqslant b \geqslant c \geqslant d \geqslant e$, $a+b+c+d+e = 100$, $b+d \leqslant a+c$, требуется найти $min(a+c+e)$. Данная задача сводится к поиску $max(b+d)$. Так как $b+d \leqslant a+c$, а $a+b+c+d+e = 100$, то $b+d \leqslant 100/2 = 50$. Такой случай легко придумать: $a, b, c, d, e = 25, 25, 25, 25, 0$. Таким образом, $min(a+c+e) = 100 - max(b+d) = 50$.

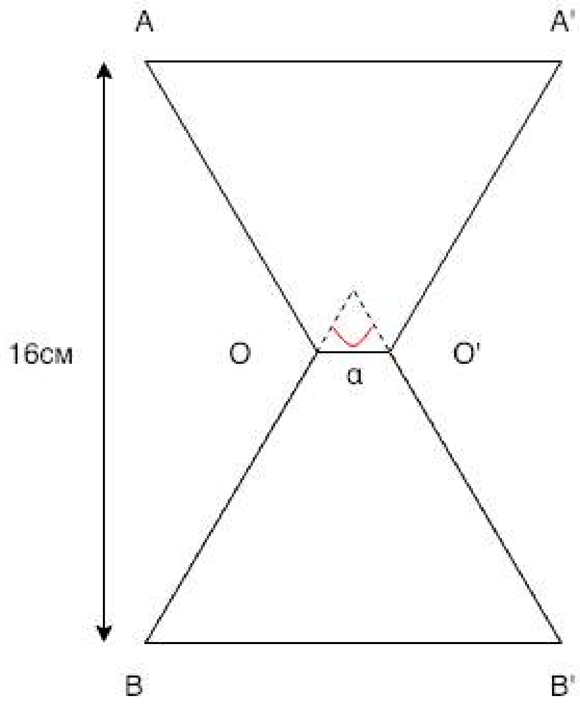
Ответ: 50.

Задача 1.6.10. (10 баллов)

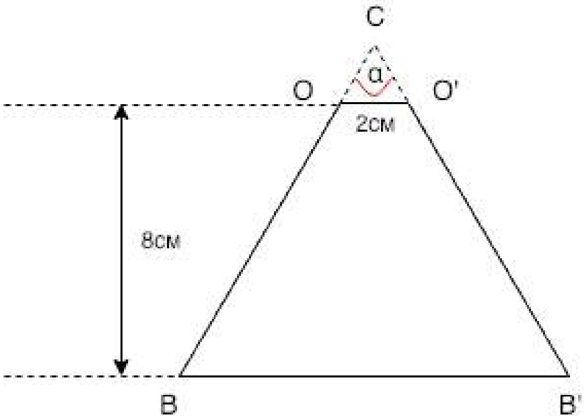
На столе стоят песочные часы высоты 16 см, представляющие собой два соединенных усеченных одинаковых конуса. Радиус горлышка (отверстия, через которое сыпется песок) равен 1 см. Тангенс угла раствора конусов равен 4/3. Чему равен объем песочных часов в см^3?

Ответ округлите до ближайшего целого.

Решение



Рассмотрим один из конусов.

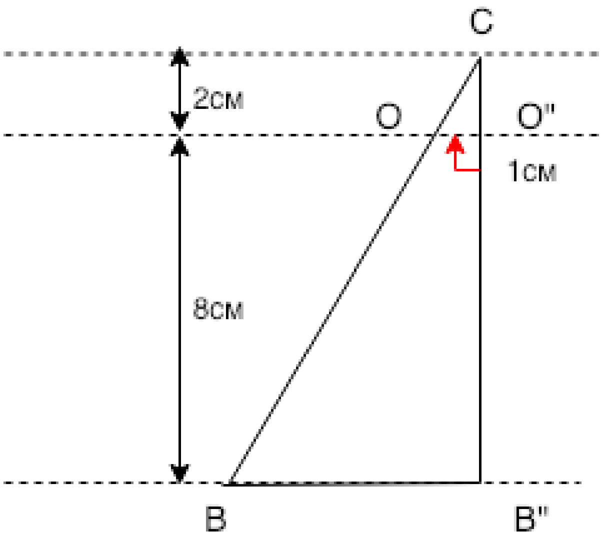


$$\operatorname{tg}^2\alpha + 1 = \frac{1}{cos^2\alpha},$$
$$cos\alpha = \frac{3}{5}.$$

Пусть $OC = O'C = x$, по теореме косинусов: $OO'^2 = x^2 + x^2 - 2x \cdot x\cos\alpha = \frac{4x^2}{5}$, $x = \sqrt{5}$.

Опустим перпендикуляр CO'' к OO''.

$$CO'' = \sqrt{x^2 - OO''^2} = \sqrt{5-1} = 2$$



Из подобия треугольников $BB'' = 5$. Объем песочных часов, т.е. удвоенный объем усеченного конуса, равен

$$V = \frac{2}{3}\pi O''B''(OO''^2 + OO''BB'' + BB''^2) = \frac{2\pi}{3} \cdot 8 \cdot (1+5+25) = 519.409 \approx 519.$$

Ответ: 519.

Задачи первого этапа. Углубленная информатика.

2.1. Первая попытка.

Задача 2.1.1. Башня (15 баллов)

Девочка Аня построила на пляже песчаный замок и хочет украсить главную башню замка лентой следующим образом:

Она хочет сделать k витков вокруг башни, а оставшуюся часть ленты разделить на ленточки равной длины и прикрепить получившиеся ленточки к вершине башни.

Аня посчитает башню красивой, если она сможет сделать желаемые k витков, затем разделить оставшуюся часть ленты на ленточки равной длины, а потом прикрепить к вершине хотя бы одну ленточку с длиной, равной высоте башни.

Родители помогли измерить Ане высоту башни, длину одного витка и длину имеющейся у неё ленты. Но Аня ещё не научилась считать, поэтому просит вас помочь определить, сможет ли она сделать свою башню красивой, используя имеющуюся у неё ленту.

Формат входных данных

В первой и последней строке четыре целых числа:

$1 \leqslant n \leqslant 1000$ – длина ленты;

$1 \leqslant m \leqslant 1000$ – длина одного витка;

$1 \leqslant k \leqslant 1000$ – количество витков;

$1 \leqslant h \leqslant 1000$ – высота башни.

Формат выходных данных

Выведите Yes, если лента подходит, иначе – No.

Примеры

Пример №1

Стандартный ввод
13 2 3 7

Стандартный вывод
Yes

Решение

Суммарная длина витков равна $k \cdot m$.

Если длины ленты не хватает даже для витков ($n < k \cdot m$), то сделать башню красивой точно не получится.

Если длины ленты хватает только на то, чтобы сделать эти k витков ($n = k \cdot m$), то сделать башню красивой не получится, так как Аня не сможет прикрепить к вершине башни ни одной ленты.

Если же можно сделать желаемые k витков и от ленты ещё что-то останется ($n > k \cdot m$), на ленточки останется ($n - k \cdot m$) единиц длины. Далее необходимо оставшуюся часть ленты разделить на ленточки равной длины и прикрепить получившиеся ленточки к вершине башни. Причём длина хотя бы одной из ленточек должна быть равна высоте башни. Следовательно длина всех ленточек должна быть равна высоте башни. А чтобы было возможно оставшуюся часть ленты разделить на ленточки с длиной, равной высоте башни, длина оставшейся части ленты должна нацело делиться на высоту башни.

Таким образом, нужно было вывести Yes, если ($n - k \cdot m$) положительно и делится нацело на h, иначе – No.

Пример программы-решения

Ниже представлено решение на языке Python3

```
n, m, k, h = [int(i) for i in input().split()]
print("Yes" if n - m * k > 0 and (n - m * k) % h == 0 else "No")
```

Задача 2.1.2. Классики (15 баллов)

Шёл 2048-ой год. Робот Вася шёл по улице и увидел играющих детей. Они прыгали то на одной ноге, то на двух. Вася подошёл и спросил, во что играют дети. Ему объяснили, что эта игра называется классики и дети прыгают по написанным на асфальте числам. Вася пропрыгал разок и пошёл дальше по своим делам.

Через несколько дней идя по улице он увидел написанные на асфальте ряды чисел и задался вопросом, классики ли это. Ещё через пару дней он снова увидел написанные на асфальте ряды чисел и вновь задался вопросом, классики ли это. Но ответить на этот вопрос он не мог.

Классиками в понимании Васи (ну, так ему объяснили) являются ряды, которые обладают следующим свойством:

Каждое целое число от 1 до $s_1 + s_2 + \ldots + s_i$ встречается в первых i рядах ровно один раз, где s_i – количество чисел в i-ом ряду.

Напишите для Васи программу, которая будет помогать ему определять, являются ли те или иные ряды чисел классиками (в понимании Васи).

Формат входных данных

В первой строке одно целое число:

$1 \leqslant t \leqslant 500$ – количество раз, которое Вася встречал написанные на асфальте ряды чисел в последнее время.

Каждый набор рядов описан следующим образом:

В первой строке одно целое число:

$1 \leqslant n \leqslant 500$ – количество рядов.

Во второй строке n целых чисел:

s_i – количество чисел в i-ом ряду

$(1 \leqslant i \leqslant n; 1 \leqslant s_i \leqslant 500)$.

Далее идёт ещё n строк. В $(i + 2)$-ой строке s_i целых чисел:

$a_{i,j}$ – j-ое число в i-ом ряду

$(|a_{i,j}| \leqslant 10^9)$.

Гарантируется, что суммарное количество чисел во входных данных не превышает 10^5.

Формат выходных данных

Для каждого набора рядов в отдельной строке выведите:

Yes, если данные ряды чисел являются классиками в понимании Васи, иначе – No.

Примеры

Пример №1

Стандартный ввод
3
5
1 2 1 2 1
1
2 3
4
5 6
7
5
1 2 1 2 1
1
3 2
4
5 6
7
5
1 2 1 2 1
1
3 2
4
6 6
7

Стандартный вывод
Yes
Yes
No

Решение

Поработаем с данным в условии задачи свойством. Какие числа находятся в i-ом ряду?

Если $i = 1$, то – числа от 1 до s_1.

Если $i > 1$, то – числа от $s_1 + s_2 + \ldots + s_{i-1} + 1$ до $s_1 + s_2 + \ldots + s_i$, так как числа от 1 до $s_1 + s_2 + \ldots + s_{i-1}$ записаны в первых $(i - 1)$ рядах.

Таким образом, для нахождения ответа для отдельно взятого набора рядов можно было, например, отсортировать числа в каждом ряду и проверить, что j-ое (после сортировки) число в i-ом ряду равно $s_1 + s_2 + \ldots + s_{i-1} + j$ для всех возможных пар (i, j).

Пример программы-решения

Ниже представлено решение на языке Python3

```
def solve():
    n = int(input())
    s = [int(i) for i in input().split()]

    ok = True
    sum = 0
    for i in range(n):
```

```
        ai = sorted([int(i) for i in input().split()])
        for j in range(s[i]):
            ok &= (ai[j] == sum + j + 1)
        sum += s[i]

    return "Yes" if ok else "No"


t = int(input())

for i in range(t):
    print(solve())
```

Задача 2.1.3. Треугольник (20 баллов)

Плоскость задана тремя точками. Определите площадь треугольника, образованного этими тремя точками.

Формат входных данных

Входные данные состоят из трёх строк. В каждой из трёх строк по три целых числа:

x_i y_i z_i – координаты i-ой точки

$(1 \leqslant i \leqslant 3; |x_i| \leqslant 10^9; |y_i| \leqslant 10^9; |z_i| \leqslant 10^9)$.

Гарантируется, что данные три точки не лежат на одной прямой.

Формат выходных данных

Одно положительное число – площадь треугольника.

Ваш ответ будет засчитан, если его абсолютная или относительная ошибка не превосходит 10^{-6}. Формально, пусть ваш ответ равен a, а ответ жюри равен b. Ваш ответ будет засчитан, если $\frac{|a-b|}{\max(1,b)} \leqslant 10^{-6}$.

Примеры

Пример №1

Стандартный ввод

`0 4 0`
`0 0 0`
`3 0 0`

Стандартный вывод
`6.0000000000`

Решение

Для нахождения площади треугольника можно, например, воспользоваться формулой Герона, которая позволяет вычислить площадь треугольника по длинам его сторон. Но точности вычислений, которой позволяют достичь стандартные типы данных для работы с действительными числами, не достаточно для прохождения всех тестов. Чтобы повысить точность, нужно либо воспользоваться типом данных для работы с длинными действительными числами, либо написать свой вариант работы с длинными действительными числами.

Пример программы-решения

Ниже представлено решение на языке Python3

```
from decimal import Decimal

x1, y1, z1 = [Decimal(i) for i in input().split()]
x2, y2, z2 = [Decimal(i) for i in input().split()]
x3, y3, z3 = [Decimal(i) for i in input().split()]
a = ((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1) + (z2 - z1) * (z2 - z1)).sqrt()
b = ((x3 - x2) * (x3 - x2) + (y3 - y2) * (y3 - y2) + (z3 - z2) * (z3 - z2)).sqrt()
c = ((x3 - x1) * (x3 - x1) + (y3 - y1) * (y3 - y1) + (z3 - z1) * (z3 - z1)).sqrt()
p = (a + b + c) / 2
print((p * (p - a) * (p - b) * (p - c)).sqrt())
```

Задача 2.1.4. R2D2 (25 баллов)

Робот R2D2 случайно оказался на Имперском корабле. Он хочет покинуть его как можно скорее. Для этого ему надо добраться до спасательной капсулы.

Для упрощения задачи корабль представляет собой прямоугольную таблицу высотой n и шириной m. Ячейка может быть либо пустой, либо представлять собой препятствие. Помогите за минимальное время добраться R2D2 из своей начальной точки до спасательной капсулы.

При этом известно, что робот может передвигаться только в клетки, соседние по стороне. То есть двигаться только вверх, вниз, влево и вправо. Также у робота есть текущее направление.

Движение вперед занимает у робота 1 секунду и поворот на 90° также занимает 1 секунду.

Зная начальное расположение робота и его направление. Выясните за какое минимальное время он сможет покинуть корабль. При этом, если робот оказался в ячейке со спасательной капсулой, его текущее направление не имеет значения.

Изначально робот всегда смотрит вниз.

Формат входных данных

В первой строке вводятся два целых числа n и m $(1 \leqslant n, m \leqslant 1000)$ – высота и ширина.

В следующих n строках вводятся m символов $a_{i,j}$. Значения ячейки $a_{i,j}$ могут быть $\#$ – препятствие, . – пустая клетка, s – начальная позиция робота, f – спасательная капсула.

Гарантируется, что ровно одна клетка в таблице имеет значение s.

Гарантируется, что ровно одна клетка в таблице имеет значение f.

Формат выходных данных

Выведите минимальное количество секунд, нужное чтобы добраться роботу до спасательной капсулы или -1, если это сделать невозможно.

Примеры

Пример №1

Стандартный ввод
3 3 s.. ... ..f

Стандартный вывод
5

Пример №2

Стандартный ввод
3 3 s.. ### ..f

Стандартный вывод
-1

Пример №3

Стандартный ввод
3 3 s.. ..f ...

Стандартный вывод
4

Решение

Давайте перейдем к графам. Вершиной будем считать пару (клетка, направление).

Ребра будут двух типов:

- перейти в соседнюю клетку в соответствии с текущим направлением;
- повернуться на $90°$, то есть сменить направление.

Тогда задача сводится к нахождению кратчайшего расстояния в получившемся графе. Задачу можно решить с помощью алгоритма поиска в ширину (BFS), так как вес всех ребер одинаков и равен 1.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
n, m = map(int, input().split())

s = []
for i in range(n):
    s.append(input())

dist = [[[2000000000 for j in range(4)] for i in range(m)] for k in range(n)]
qi = [0] * (n * m * 4)
qj = [0] * (n * m * 4)
qd = [0] * (n * m * 4)
ql = 0
qr = 0

for i in range(n):
    for j in range(m):
        if (s[i][j] == 's'):
            dist[i][j][0] = 0
            qi[qr] = i
            qj[qr] = j
            qd[qr] = 0
            qr += 1
            break
    if qr > 0:
        break

ans = -1
while ql < qr:
    i = qi[ql]
    j = qj[ql]
    d = qd[ql]
    ql += 1

    if s[i][j] == 'f':
        ans = dist[i][j][d]
        break

    if d == 0:
        if i + 1 < n and s[i + 1][j] != '#' and dist[i + 1][j][d] > dist[i][j][d] + 1:
            dist[i + 1][j][d] = dist[i][j][d] + 1
            qi[qr] = i + 1
            qj[qr] = j
            qd[qr] = d
            qr += 1
    elif d == 1:
        if j > 0 and s[i][j - 1] != '#' and dist[i][j - 1][d] > dist[i][j][d] + 1:
            dist[i][j - 1][d] = dist[i][j][d] + 1
            qi[qr] = i
            qj[qr] = j - 1
            qd[qr] = d
            qr += 1
    elif d == 2:
        if i > 0 and s[i - 1][j] != '#' and dist[i - 1][j][d] > dist[i][j][d] + 1:
            dist[i - 1][j][d] = dist[i][j][d] + 1
            qi[qr] = i - 1
            qj[qr] = j
            qd[qr] = d
```

```
            qr += 1
    elif d == 3:
        if j + 1 < m and s[i][j + 1] != '#' and dist[i][j + 1][d] > dist[i][j][d] + 1:
            dist[i][j + 1][d] = dist[i][j][d] + 1
            qi[qr] = i
            qj[qr] = j + 1
            qd[qr] = d
            qr += 1

    if dist[i][j][(d + 1) % 4] > dist[i][j][d] + 1:
        dist[i][j][(d + 1) % 4] = dist[i][j][d] + 1
        qi[qr] = i
        qj[qr] = j
        qd[qr] = (d + 1) % 4
        qr += 1

    if dist[i][j][(d + 3) % 4] > dist[i][j][d] + 1:
        dist[i][j][(d + 3) % 4] = dist[i][j][d] + 1
        qi[qr] = i
        qj[qr] = j
        qd[qr] = (d + 3) % 4
        qr += 1

print(ans)
```

Задача 2.1.5. Робот и камушки (25 баллов)

Ильнар в одной из комнат увидел странного робота. Во время выполнения алгоритма, он доставал из мешка разные камушки. Причем он никогда не доставал один и тот же камень два раза. И говорил сколько камушков в мешке такого же цвета.

Разработчик робота рассказал Ильнару, что из-за ошибки в коде робот ровно один раз всегда ошибается.

Теперь Ильнару интересно, а сколько минимально может быть камушков в мешке.

Примечание

Робот говорит количество камушков того же цвета, что в руке. При этом камень, который у него в руках, он тоже учитывает. После данной операции камень возвращается обратно в мешок.

Формат входных данных

В первой строке содержится единственное целое число n $(1 \leqslant n \leqslant 10^5)$ – количество выбранных камушков.

Во второй строке находятся n целых чисел a_i $(1 \leqslant a_i \leqslant 10^9)$ – значения, названные роботом.

Формат выходных данных

Выведите одно положительное целое число – минимальное возможное количество камушков в мешке.

Примеры

Пример №1

Стандартный ввод
4 2 2 2 2
Стандартный вывод
5

Решение

Сгруппируем все одинаковые значения.

Пусть cnt_x – количество камней, про которые сказано, что такого же цвета x камней.

Если бы все n высказываний были правдивы, ответом было бы $\sum\limits_{x \in \mathbb{N}} ceil(\frac{cnt_x}{x}) \cdot x$, где ceil – округление вверх.

Так как одно высказывание ложно, то некоторое значение cnt_{x_1} должно увеличиться на 1, а некоторое cnt_{x_2} должно уменьшиться на 1.

Давайте для каждого значения x посчитаем, как изменится ответ, если мы прибавим 1 к cnt_x и если отнимем 1 от cnt_x.

Дальше нужно найти такую пару, которая минимизирует суммарный результат. Это сделать можно, например, с помощью динамического программирования.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
n = int(input())

a = list(map(int, input().split()))
a.sort()

sum = 0
hide = []
hide.append(1)
toHide = 0
l = 0
r = 0

while l < n:
    while r < n and a[l] == a[r]:
        r += 1
    sum += (r - l + a[l] - 1) // a[l] * a[l]
    if (r - l + a[l] - 2) // a[l] < (r - l + a[l] - 1) // a[l]:
        toHide = a[l]
    if (r - l + a[l]) // a[l] == (r - l + a[l] - 1) // a[l]:
        hide.append(a[l])
    l = r

```

```
hide.reverse()

if toHide == 0:
    if len(hide) == 1 or (a[0] == hide[0] and a[n - 1] == hide[0]):
        sum += 1
elif toHide == 1:
    if len(hide) > 1:
        sum -= 1
    else:
        sum += 1
else:
    for i in hide:
        if i != toHide:
            sum = sum - toHide + (1 if i == 1 else 0)
            break

print(sum)
```

2.2. Вторая попытка.

Задача 2.2.1. Призы (10 баллов)

Организаторы олимпиады по робототехнике решили, что у них будет ровно n участников финального этапа. Они хотят подарить каждому участнику батончики KitKat и уже нашли, что можно купить по низкой цене пачки, в которых будет ровно по m батончиков.

Организаторы решили, что все участники должны получить поровну батончиков, иначе кто-то обидится. Также они решили, что не должно остаться лишних батончиков. Поэтому им нужно определить минимальное количество пачек, которые они должны купить.

Формат входных данных

Вводится два целых числа n и m ($1 \leqslant n, m \leqslant 10^9$) – количество участников и количество батончиков в пачке.

Формат выходных данных

Выведите единственное число – ответ на задачу.

Примеры

Пример №1

Стандартный ввод
8 4

Стандартный вывод
2

Решение

Задача сводится к тому, что нужно найти такое минимальное число k, что $k \cdot m$ делится на n без остатка.

Получается, что $k \cdot m = lcm(m, n)$, где lcm – это наименьшее общее кратное.

Так как $lcm(a, b) \cdot gcd(a, b) = a \cdot b$, где lcm – это наименьшее общее кратное, а gcd – это наибольший общий делитель, получаем:

$lcm(m, n) \cdot gcd(m, n) = m \cdot n \Rightarrow k \cdot m = lcm(m, n) = \frac{m \cdot n}{gcd(m,n)} \Rightarrow k = \frac{n}{gcd(m,n)}$

Таким образом, ответом будет $\frac{n}{gcd(m,n)}$.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
from fractions import gcd

n, m = [int(i) for i in input().split()]
print(n // gcd(n, m))
```

Задача 2.2.2. k-экстремум (15 баллов)

Дан массив a из n элементов.

Элемент массива a_i является k-экстремумом, если он является k-максимумом или k-минимумом.

Элемент массива a_i $(1 < i < n)$ является k-максимумом тогда и только тогда, когда $a_{i-1} + k \leqslant a_i$ и $a_{i+1} + k \leqslant a_i$.

Элемент массива a_i $(1 < i < n)$ является k-минимумом тогда и только тогда, когда $a_{i-1} - k \geqslant a_i$ и $a_{i+1} - k \geqslant a_i$.

Если a_i – один из крайних ($i = 1$ или $i = n$) элементов массива, то он не может быть k-экстремумом.

За одну единицу времени вы можете увеличить или уменьшить на единицу любой элемент массива.

Какое минимальное количество времени нужно потратить, чтобы в массиве появился хотя бы один k-экстремум?

Формат входных данных

В первой строке вводятся два целых числа n и k $(3 \leqslant n \leqslant 10^5, 1 \leqslant k \leqslant 10^9)$.

Во второй строке вводятся n чисел a_i $(1 \leqslant a_i \leqslant 10^9)$.

Формат выходных данных

Выведите единственное число – ответ на задачу.

Примеры

Пример №1

Стандартный ввод
3 2 4 3 4

Стандартный вывод
1

Решение

Заметим, что тратить время стоит только на увеличение или только на уменьшение одного конкретного элемента массива a_i $(1 < i < n)$.

Давайте для каждого элемента посчитаем, сколько нужно потратить времени, чтобы сделать его k-экстремумом.

Чтобы сделать i-ый $(1 < i < n)$ элемент k-максимумом, его значение должно стать не менее $\max(a_{i-1}+k, a_{i+1}+k)$, а на это нужно потратить $\max(a_i, a_{i-1}+k, a_{i+1}+k) - a_i$ единиц времени.

Чтобы сделать i-ый $(1 < i < n)$ элемент k-минимумом, его значение должно стать не более $\min(a_{i-1}-k, a_{i+1}-k)$, а на это нужно потратить $a_i - \min(a_i, a_{i-1}-k, a_{i+1}-k)$ единиц времени.

Тогда время, которое необходимо потратить, чтобы сделать i-ый элемент k-экстремумом, будет равно минимуму из этих двух значений. А ответом на всю задачу – минимальное значение по всем элементам.

Пример программы-решения

Ниже представлено решение на языке C++

```cpp
#include <bits/stdc++.h>

using namespace std;

const int MAXN = 1e5 + 1;
int a[MAXN];

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    int n;
    cin >> n;
    int k;
    cin >> k;
    int ans = 1e9 + 500;
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
    }
    for (int i = 2; i <= n - 1; ++i) {
```

```
        //bigger
        int val = a[i];
        val = max(val, a[i - 1] + k);
        val = max(val, a[i + 1] + k);
        ans = min(ans, val - a[i]);
        //smaller
        val = a[i];
        val = min(val, a[i - 1] - k);
        val = min(val, a[i + 1] - k);
        ans = min(ans, a[i] - val);

    }
    cout << ans << endl;

    return 0;
}
```

Задача 2.2.3. Шпионаж (20 баллов)

Наше агентство осуществило перехват нескольких предположительно шпионских сообщений. Однако возникли проблемы при декодировании.

Нам удалось узнать, что:

- каждый символ изначального сообщения закодировали последовательностью из нулей и единиц;
- длина каждой из этих последовательностей равна k ;
- каждому символу поставлена в соответствие ровно одна последовательность из k нулей и единиц;
- каждой последовательности из k нулей и единиц поставлен в соответствие ровно один символ;
- экземпляры таблицы декодирования испорчены и не подлежат восстановлению.

Большего вам знать не нужно.

Для первичного отделения шпионских сообщений от сообщений, попавших в рассмотрение случайно, нам нужна программа, подсчитывающая количество различных символов, используемых в сообщении, представленном в виде строки.

Берётесь за эту работу?

Формат входных данных

В первой строке входных данных два целых числа:

$1 \leqslant n \leqslant 10^5$ – длина строки;

$1 \leqslant k \leqslant n$ – длина последовательностей, которыми были закодированы символы.

Во второй строке дано сообщение в виде строки s .

Гарантируется, что число n кратно k и закодированная строка s состоит из n символов, каждый из которых равен 0 или 1 .

Формат выходных данных

Выведите одно положительное число – количество различных символов в строке.

Примеры

Пример №1

Стандартный ввод
9 3 001000100

Стандартный вывод
3

Решение

В данной задаче нужно было разбить данную строку на строки длины k и определить количество различных строк в получившемся наборе строк.

Для этого можно было, например, отсортировать строки и посчитать количество строк, каждая из которых не равна предшествующей ей строке в получившемся отсортированном массиве строк. А можно было добавить все строки набора в множество (set) и посмотреть на размер получившегося множества.

Пример программы-решения

Ниже представлено решение на языке Python3

```
n, k = (int(i) for i in input().split())
s = input()
print(len(set(s[i * k : (i + 1) * k] for i in range(n // k))))
```

Задача 2.2.4. Цепь (25 баллов)

Герои книги "Автостопом по галактике" в своем путешествии случайно попали в параллельную вселенную. В этой вселенной все по-другому. Можно перемещаться не только по пространству, но и во времени. Для этого они используют свой звездолет.

Законы физики и всех других наук здесь работают не так.

В этой вселенной есть n планет, для каждой планеты известно ее расстояние от центра вселенной r_i и коэффициент искажения пространства k_i.

Чтобы сделать эту вселенную более интересной для туристов из других вселенных, наши герои решили соединить все планеты с помощью пар звездных врат. Между звездными вратами можно перемещаться в обе стороны. При этом для каждой планеты герои выбрали год y_i, когда эта планета наиболее интересна туристам. Герои хотят, чтобы после этого построения звездных врат можно было посетить все планеты.

При этом для построения звездных врат нужно будет потратить часть энергии звездолета.

Чтобы построить портал между i-ой и j-ой планетами, нужно потратить

$3 \cdot (r_i - r_j)^2 + 2 \cdot |(2 \cdot k_i - 2 \cdot k_j) \cdot (2 \cdot k_i + 2 \cdot k_j)| + 5 \cdot |y_i - y_j|$ единиц энергии.

Герои хотят минимизировать суммарное количество энергии, которое нужно потратить на построение звездных врат, чтобы как можно больше топлива осталось им для остальных путешествий. Помогите им это сделать.

Формат входных данных

В первой строке вводится целое число n ($1 \leqslant n \leqslant 10^4$) – количество планет.

В следующих n строках вводятся по три целых числа r_i, k_i, y_i ($0 \leqslant r_i, k_i, y_i \leqslant 10^6$)
.

Формат выходных данных

Выведите единственное число – минимальное суммарное количество энергии.

Примеры

Пример №1

Стандартный ввод
2 1 1 1 2 2 2

Стандартный вывод
32

Решение

Переведём задачу на язык графов: планеты – это вершины графа, а звездные врата между планетами – рёбра.

Необходимо обеспечить существование пути между любой парой вершин в графе, причём суммарный вес всех рёбер в графе должен быть минимален. То есть задача сводится к нахождению минимального остовного дерева.

Поскольку в задаче дан полный граф с достаточно большим количеством вершин, для решения данной задачи выгоднее выбрать алгоритм Прима для плотных графов с асимптотикой работы $O(n^2 + m)$, нежели алгоритм Краскала или алгоритм Прима для разреженных графов.

Пример программы-решения

Ниже представлено решение на языке C++

```cpp
#include <iostream>

using namespace std;
typedef long long ll;

```

```
const int MAXN = 10000 + 1;
const ll inf = 1e18;
ll r[MAXN], k[MAXN], y[MAXN];

inline ll my_abs(ll x) {
    if (x > 0)
        return x;
    return -x;
}

inline ll cost(int i, int j) {
    return 3 * (r[i] - r[j]) * (r[i] - r[j]) +
           2 * my_abs(2 * k[i] - 2 * k[j]) * my_abs(2 * k[i] + 2 * k[j]) +
           5 * my_abs(y[i] - y[j]);
}

ll mn[MAXN];
bool used[MAXN];

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    int n;
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> r[i] >> k[i] >> y[i];
        mn[i] = inf;
    }

    ll ans = 0;

    for (int step = 1; step <= n; ++step) {
        int ver = -1;
        for (int i = 1; i <= n; ++i) {
            if (used[i])
                continue;
            if (ver == -1 || mn[i] < mn[ver]) {
                ver = i;
            }
        }
        used[ver] = 1;
        if (step > 1)
            ans += mn[ver];
        for (int j = 1; j <= n; ++j) {
            if (used[j])
                continue;
            mn[j] = min(mn[j], cost(ver, j));
        }

    }
    cout << ans << endl;
    return 0;
}
```

Задача 2.2.5. Сжали (30 баллов)

Строка t является результатом сжатия строки s, если (должны выполняться все пункты):

- Строка t является подпоследовательностью строки s .

  Строка t длины k является подпоследовательностью строки s длины n, если существует такой набор чисел p, что i-ый символ строки t равен p_i-ому символу строки s и выполняется неравенство $1 \leqslant p_1 < p_2 < \ldots < p_{k-1} < p_k \leqslant n$.
- Первый символ строки t равен первому символу строки s (то есть $p_1 = 1$).
- Последний символ строки t равен последнему символу строки s (то есть $p_k = n$).
- Если i-ый ($1 \leqslant i < k$) символ строки t является m-ой буквой латинского алфавита, то в удовлетворяющем строке t наборе чисел p между p_i-ым и p_{i+1}-ым символами в строке s не больше $bonus_m$ символов (то есть $p_{i+1} - p_i - 1 \leqslant bonus_m$) .

Стоимость строки определяется как сумма стоимостей всех ее символов. Стоимость i-ой буквы латинского алфавита равна $cost_i$.

Определите стоимость самой дешёвой из сжатых строк s .

Формат входных данных

В первой строке входных данных одно целое число:

$3 \leqslant n \leqslant 10^6$ – длина строки s .

Во второй строке – сама строка s , состоящая из строчных латинских букв.

В третьей строке – 26 целых чисел:

$0 \leqslant cost_i \leqslant 10^9$ – стоимость добавления в сжатую строку i-го ($1 \leqslant i \leqslant 26$) символа латинского алфавита.

В четвертой строке – 26 целых чисел:

$0 \leqslant bonus_i \leqslant 10^9$ – сколько символов можно пропустить в строке s после добавления в сжатую строку i-го ($1 \leqslant i \leqslant 26$) символа латинского алфавита.

Формат выходных данных

Выведите одно неотрицательное число – стоимость самой дешёвой из полученных описанным сжатием строк.

Примеры

Пример №1

Стандартный ввод
8 abcdebha 0 2 2 3 0 0 1 1 2 1 0 3 2 0 2 2 3 1 0 1 2 2 0 1 2 3 1 2 0 2 1 1 2 0 1 1 1 1 0 1 2 0 0 1 2 2 3 0 1 3 1 2

Стандартный вывод
3

Решение

У данной задачи множество различных решений. Перечислим некоторые из них:

- Решение с использованием динамического программирования и дерева отрезков.
  Асимптотика: $O(n \cdot \log(n))$
- Решение с использованием динамического программирования и множества (set).
  Асимптотика: $O(n \cdot \log(n))$
- Решение с использованием динамического программирования.
  Асимптотика: $O(n \cdot 26)$
- Решение с использованием динамического программирования и стека на массиве.
  Асимптотика: $O(n \cdot \log(n))$

Мы подробно разберём только одно решение – решение с использованием динамического программирования и стека на массиве. И начнём с части про динамическое программирование, которая присутствует в каждом решении в том или ином виде.

Ответ для префикса строки длины i равен:

$$dp_i = \begin{cases} cost_{s_i}, & \text{если } i = 1; \\ min(dp_j) + cost_{s_i}, & \text{если } i > 1\text{; ограничения на } j\text{: } 1 \leqslant j < i \text{ и } i - j - 1 \leqslant bonus_{s_j}. \end{cases}$$

Тогда ответом на задачу является dp_n.

Посчитать такую динамику можно за $O(n \cdot \log(n))$, если умело воспользоваться множеством (set), но мы пойдём другим путём.

Развернём строку. Теперь ответ для префикса (который на самом деле является суффиксом) строки длины i равен:

$$dp_i = \begin{cases} cost_{s_i}, & \text{если } i = 1; \\ min(dp_j) + cost_{s_i}, & \text{если } i > 1\text{; ограничения на } j\text{: } \max(1, i - 1 - bonus_{s_i}) \leqslant j < i. \end{cases}$$

Ответом на задачу по-прежнему является dp_n.

Такую динамику достаточно быстро (за $O(n \cdot \log(n))$ или $O(n \cdot 26)$) можно посчитать разными способами. Мы для нахождения dp_n воспользуемся стеком на массиве.

Будем поддерживать стек на массиве: первый элемент массива – дно стека, а последний элемент – вершина стека. На элементы стека будет наложено следующее условие:

Если i лежит в стеке и лежит не на дне, то предшествующий ему в стеке элемент – это такое j, что $dp_j < dp_i$ и $j < i$; причём из всех j, удовлетворяющих этим условиям, предшествует i в стеке наибольшее j. Если i лежит на дне стека, то такого j, что $dp_j < dp_i$ и $j < i$, не существует.

Добавлять элементы в стек будем в порядке возрастания i. При $i = 1$ стек пуст и мы просто добавляем в стек i. При $i > 1$ сначала удалим с вершины стека элементы, которым соответствуют значения динамики, которые не меньше dp_i, и только потом добавим в стек i. Таким образом мы получаем новый стек, который не содержит ни одного элемента, относительно которого не соблюдалось бы описанное выше условие, и на вершине стека лежит i.

Для вычисления dp_i $(i > 1)$ воспользуемся стеком, полученным описанным способом и на вершине которого лежит $(i - 1)$. Не сложно заметить, что j, при котором достигается значение $min(dp_j)$ в рекуррентной формуле для dp_i, равен наименьшему элементу стека, который не меньше $\max(1, i-1-bonus_{s_i})$. Поскольку элементы стека строго возрастают от дна стека к его вершине, такой элемент можно найти бинарным поиском (за $O(\log(n))$).

Так мы вычислим значение dp_n за $O(n \cdot \log(n))$.

Пример программы-решения

Ниже представлено решение на языке Python3

```
n = int(input())

s = input().strip()
s = s[::-1]

alphabet = 26
cost = list(map(int, input().split()))
bonus = list(map(int, input().split()))

ans = [0] * n
ans[0] = cost[ord(s[0]) - ord('a')]

st = [0] * n
ss = 1

for i in range(n - 1):
    l = -1
    r = ss - 1
    while l + 1 < r:
        mid = (l + r) // 2
        if st[mid] < i - bonus[ord(s[i + 1]) - ord('a')]:
            l = mid
        else:
            r = mid

    ans[i + 1] = ans[st[r]] + cost[ord(s[i + 1]) - ord('a')]

    while ss > 0 and ans[st[ss - 1]] >= ans[i + 1]:
```

```
            ss -= 1
        st[ss] = i + 1
        ss += 1

print(ans[n - 1])
```

2.3. Третья попытка.

Задача 2.3.1. Печеньки (10 баллов)

У Ильнара есть три коробки с печеньем, в которых a, b, c печенек соответственно. К Ильнару в гости пришло $(n - 1)$ человек. Ильнар хочет, чтобы всем гостям и ему досталось одинаковое количество печенек. Поэтому он хочет узнать сколько печенек достанется каждому, если он откроет несколько (возможно ни одной, возможно все три) коробок с печеньем, при этом все печеньки из открытых коробок должны быть розданы поровну (сами печенья нельзя ломать на части).

Определите максимальное количество печенек, которое получит каждый гость и Ильнар.

Формат входных данных

В первой строке вводится 4 целых числа a, b, c, n $(1 \leqslant a, b, c, n \leqslant 10^8)$.

Формат выходных данных

Выведите единственное число – ответ на задачу.

Примеры

Пример №1

Стандартный ввод
1 2 3 4

Стандартный вывод
1

Пример №2

Стандартный ввод
3 4 5 19

Стандартный вывод
0

Решение

Переберем все возможные варианты выбора коробок. Получаем 7 вариантов количества печенек в открытых коробках:

$a, b, c, a+b, a+c, b+c, a+b+c$

Из них возьмем максимальное значение, которое делится на n без остатка. Если все 7 значений не делятся на n без остатка, то ответ – 0.

Пример программы-решения

Ниже представлено решение на языке Python3

```python
a, b, c, n = [int(x) for x in input().split()]
for i in sorted([a, b, c, a + b, a + c, b + c, a + b + c, 0], reverse=True):
    if i % n == 0:
        print(i // n)
        break
```

Задача 2.3.2. Круглый стол (20 баллов)

Организаторы одного из турниров по одной из самых известных карточных игр заказали круглый стол, за которым будет сидеть несколько (минимум 2) игроков во время каждой из игр.

Профсоюз игроков этой карточной игры выдвинул условие: каждый игрок на время игры должен иметь личное пространство как минимум длины d.

Если посчитать стол кругом, то длина пространства игрока определяется как длина хорды, которая отделяет сегмент круга, на котором игрок может положить свои карты и руки, от остальной части стола.

Помогите организаторам определить, какое максимальное количество игроков может одновременно сидеть за этим столом.

Формат входных данных

Вводится два целых числа d и r $(1 \leqslant d, r \leqslant 2 \cdot 10^9)$ – минимальная длина личного пространства и радиус круглого стола.

Формат выходных данных

Если нельзя усадить за стол даже двух игроков, выведите -1, иначе выведите единственное положительное число – максимальное количество игроков, которых можно разместить за столом.

Примеры

Пример №1

Стандартный ввод
6 3

Стандартный вывод
2

Пример №2

Стандартный ввод
7 2

Стандартный вывод
-1

Пример №3

Стандартный ввод
12 1000

Стандартный вывод
523

Пример №4

Стандартный ввод
3 3

Стандартный вывод
6

Решение

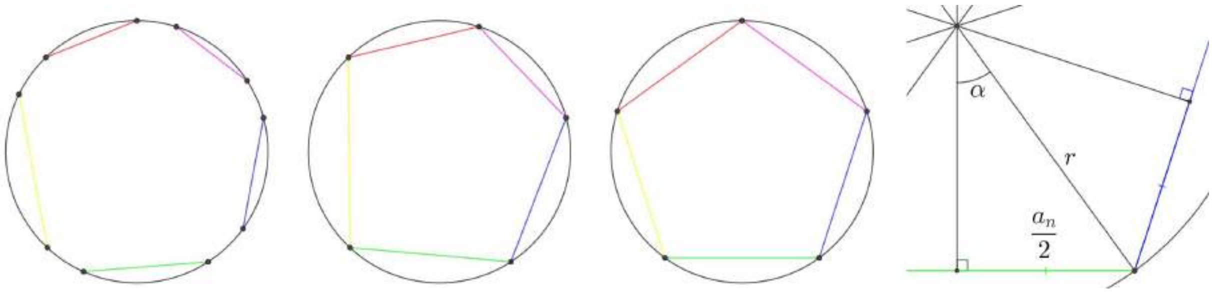
Очевидно, что если $d > 2 \cdot r$, то за стол даже двух игроков не посадить и в качестве ответа нужно выводить -1. Далее будем решать задачу с дополнительным ограничением $d \leqslant 2 \cdot r$.

Заметим, что при фиксированном количестве игроков, которых организаторы хотят усадить за стол, мы можем рассчитать наибольшую длину личного пространства, которое можно предоставить каждому из игроков.

Пусть организаторы хотят усадить за стол $n = 5$ игроков. Любым способом, который не противоречит условию, проведём хорды, отделяющие личное пространство игроков от остальной части стола (см. рис. 1).

Заметим, что могут остаться дуги, "не покрытые" ни одной из хорд. "Покроем" каждую из таких дуг одной из близлежащих хорд (см. рис. 2), что позволит не только не уменьшить длину наименьшего из предоставленных личных пространств, но и, возможно, увеличить эту длину.

Таким образом, получившиеся хорды стали образовывать n-угольник, около которого описана окружность.



Также заметим, что если существует пара хорд, длины которых различны, то длину меньшей из них можно увеличить за счёт уменьшения длины большей.

А теперь заметим, что для максимизации длины личного пространства необходимо сделать все хорды равными по длине, поскольку если длина меньшей хорды не равна длине большей, то меньшую хорду можно увеличить за счёт большей, таким образом увеличив длину личного пространства.

В итоге хорды стали образовывать правильный n-угольник (см. рис. 3).

Используя свойства правильных многоугольников, можно вывести формулу, выражающую длину стороны правильного n-угольника через радиус описанной около n-угольника окружности (см. рис. 4):

$$\sin(\alpha) = \sin(\tfrac{2\cdot\pi}{2\cdot n}) = \tfrac{a_n}{2\cdot r} \Leftrightarrow a_n = 2\cdot r\cdot \sin(\tfrac{\pi}{n})\ ,$$

где a_n – длина стороны правильного n-угольника, которая в свою очередь равна наибольшей длине личного пространства, которое мы можем предоставить каждому из n игроков на столе радиуса r.

Таким образом, остаётся определить ответ, который равен такому наибольшему n, что $a_n \geqslant d$. Искомое n можно найти бинарным поиском по ответу, так как последовательность, заданная формулой $a_i = 2\cdot r\cdot\sin(\frac{\pi}{i})$ для $i \geqslant 2$, является убывающей. Но можно пойти дальше и доказать, что ответом на задачу является $n = \left\lfloor \frac{\pi}{\arcsin(\frac{d}{2\cdot r})} \right\rfloor$

Пример программы-решения

Ниже представлено решение на языке C++

```cpp
#include <iostream>
#include <cmath>

#define EPS 1e-9

using namespace std;

int main() {
    double d, r;
    cin >> d >> r;
    double sine = d / 2 / r;
    if (sine > 1)
        cout << -1;
    else {
        double angle = asin(sine);
        cout << (long long) floor(M_PI / angle + EPS);
    }
    return 0;
}
```

Задача 2.3.3. Анна и красивая строка (20 баллов)

Анна написала генератор красивых строк. Она считает строку красивой, если она одинаково читается как слева направо, так и справа налево. Например, *rrr* и *anna* – красивые строки, а *abc* и *adba* – нет.

Но она допустила ошибку в коде и генератор выводит не красивые строки, а строки, которые можно сделать красивыми, если из каждой удалить ровно один символ. По крайней мере, она так думает.

Она просит вас помочь определить, верно ли её предположение.

Формат входных данных

В первой строке вводится строка s состоящая из маленьких латинских букв ($4 \leqslant |s| \leqslant 10^5$, где $|s|$ – длина строки).

Формат выходных данных

Выведите *YES* , если можно удалить один символ из строки так, чтобы она стала красивой; иначе – *NO* .

Примеры

Пример №1

Стандартный ввод
`abca`
Стандартный вывод
`YES`

Пример №2

Стандартный ввод
`abcd`
Стандартный вывод
`NO`

Решение

В данной задаче нас просят определить, может ли строка s длины n стать палиндромом, если из неё удалить ровно один символ. Если может, то ответ *YES* , иначе – *NO* .

Заметим, что если строка s является палиндромом, то можно удалить из неё $\left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right)$-ый символ и строка s останется палиндромом, то есть в качестве ответа нужно вывести *YES* .

Теперь рассмотрим случай, когда строка s не является палиндромом, то есть существует такое k ($k \geqslant 1$), что $s_1 = s_n$, $s_2 = s_{n-1}$, ..., $s_{k-1} = s_{n-k+2}$, $s_k \neq s_{n-k+1}$ ($k < n - k + 1$), где s_i – i-ый символ строки.

Докажем, что если $s_1 = s_n$, то ответ для строки s, из которой предварительно удалили первый и последний символы, будет такой же, как и для самой строки s.

Докажем от противного. Если утверждение не верно, то строка s может стать палиндромом после удаления i-го символа ($1 \leqslant i \leqslant n$), только если $i = 1$ или $i = n$. Но если после удаления первого символа получился палиндром, то получается, что $s_2 = s_n = s_1$, то есть не важно, удалим мы первый или второй символ, а если палиндром получился после удаления последнего символа, то $s_{n-1} = s_1 = s_n$, то есть не важно, удалим мы последний или предпоследний символ. Пришли к противоречию.

Следовательно, если после удаления одного символа строка s может стать палиндромом, то существует такое i, что соблюдается двойное неравенство $k \leqslant i \leqslant n-k+1$ и строка s станет палиндромом, если удалить i-ый символ. Также заметим, что после удаления такого i-го символа, что соблюдается двойное неравенство $k < i < n-k+1$, строка s точно не станет палиндромом, так как $s_k \neq s_{n-k+1}$, что противоречит определению палиндрома.

Таким образом, для определения ответа достаточно посмотреть, станет ли строка s палиндромом после удаления k-го символа и станет ли строка s палиндромом после удаления $(n-k+1)$-го символа. Если хотя бы в одном из этих двух случаев строка s станет палиндромом, то ответ *YES* , иначе – *NO* .

Пример программы-решения

Ниже представлено решение на языке Python3

```python
s = input().strip()
n = len(s)
for i in range((n - 1) // 2 + 1):
    if s[i] != s[n - 1 - i] or i == (n - 1) // 2:
        t = "".join([('' if n - 1 - i == j else s[j]) for j in range(n)])
        s = "".join([('' if i == j else s[j]) for j in range(n)])
        break
print('YES' if t == t[::-1] or s == s[::-1] else 'NO')
```

Задача 2.3.4. Робот (25 баллов)

Ильнар решил научить робота перемещаться по плоскости. Но у него оставалось мало времени и он решил немного схитрить. Он научил робота перемещаться правильно только на дистанцию не более чем $\sqrt{k}$. Чтобы робот смог при этом добраться от начальной точки до конечной, он задал в конфигурации еще дополнительные точки на плоскости. Теперь робот может перемещаться только между этими точками.

Чтобы добраться от точки i до точки j, робот тратит $(x_i - x_j)^2 + (y_i - y_j)^2$ единиц времени.

Формат входных данных

В первой строке заданы числа n и k $(1 \leqslant n \leqslant 5000,\ 1 \leqslant k \leqslant 10^9)$.

В следующих n строках заданы координаты точек x_i, y_i $(1 \leqslant x_i, y_i \leqslant 20000)$.

В последней строке заданы два числа s и t $(1 \leqslant s, t \leqslant n,\ s \neq t)$ – начальная и конечная точки пути робота.

Формат выходных данных

Выведите единственное число – минимальное время за которое робот доберется из s в t или -1, если робот не может добраться.

Примеры

Пример №1

Стандартный ввод
2 32 1 1 5 5 1 2

Стандартный вывод
32

Пример №2

Стандартный ввод
2 31 1 1 5 5 1 2

Стандартный вывод
-1

Решение

Переведём задачу на язык графов: точки – это вершины графа, а отрезки, соединяющие две точки, квадрат расстояния между которыми не превышает k – рёбра графа.

Задача сводится к тому, что нужно найти кратчайшее расстояние от вершины, соответствующей точке s, до вершины, соответствующей точке t, во взвешенном графе, веса рёбер которого неотрицательны. А для решения такой задачи можно воспользоваться алгоритмом Дейкстры.

Также стоит отметить, что поскольку количество вершин в графе не превышает 5000 и описанный в тесте граф может быть достаточно плотным, то стоит отдать предпочтение реализации алгоритма Дейкстры для плотных графов с асимптотикой работы $O(n^2 + m)$.

Пример программы-решения

Ниже представлено решение на языке C++

```cpp
#include <bits/stdc++.h>

using namespace std;
```

```cpp
typedef long long ll;
const ll inf_ll = 1e16;
const int MAXN = 1e5 + 100;

ll x[MAXN], y[MAXN];
ll dis[MAXN];

ll dist(int a, int b) {
    return (x[a] - x[b]) * (x[a] - x[b]) + (y[a] - y[b]) * (y[a] - y[b]);
}

bool used[MAXN];

int main() {
    int n;
    cin >> n;
    ll k;
    cin >> k;
    for (int i = 1; i <= n; ++i) {
        cin >> x[i] >> y[i];
    }
    int s, t;
    cin >> s >> t;
    for (int i = 1; i <= n; ++i) {
        dis[i] = inf_ll;
    }
    dis[s] = 0;
    for (int i = 1; i <= n; ++i) {
        int ver = -1;
        for (int e = 1; e <= n; ++e) {
            if (!used[e] && (ver == -1 || dis[e] < dis[ver])) {
                ver = e;
            }
        }

        int v = ver;
        used[v] = 1;
        for (int to = 1; to <= n; ++to) {
            if (dist(v, to) > k)
                continue;
            if (dis[to] > dis[v] + dist(v, to)) {
                dis[to] = dis[v] + dist(v, to);
            }
        }
    }

    if (dis[t] == inf_ll) {
        cout << -1 << endl;
    } else {
        cout << dis[t] << endl;
    }
}
```

Задача 2.3.5. ЗАГС (25 баллов)

В стране M есть город N. А в этом городе есть ЗАГС. И сегодня там случилась поломка ПО.

Были потеряны данные о заявлениях на регистрацию брака. Однако остался журнал прихода и вызова.

В журнале в хронологическом порядке записаны события двух видов:

- Приход человека, желающего подать заявление. В журнале записана фамилия этого человека.
- Объявление о вызове для утверждения заявления и проверки документов. В этом случае в журнале записано слово $next$.

В стране M принято, что супруги должны иметь одинаковую фамилию и начиная с прихода в ЗАГС для подачи заявления на регистрацию брака должны называть фамилию, которую будут носить после вступления в брак.

В городе N заявление на регистрацию брака принимается только в присутствии обоих будущих супругов.

После прихода в ЗАГС и внесения в журнал прихода и вызова соответствующей записи нужно встать в очередь. В случае, если один из будущих супругов пришёл раньше, то другой или другая присоединяются к будущей супруге или будущему супругу.

Когда вызывают следующую пару для утверждения заявления, может оказаться так, что первым в очереди стоит человек, будущая супруга или будущий супруг которого ещё не пришёл в ЗАГС. В этом случае на утверждение заявления идут те уже пришедшие в ЗАГС будущие супруги, которые находятся ближе всего к началу очереди. В случае, если в очереди нет ни одной пары, на вызов пойдёт первая появившаяся в очереди пара сразу после прихода второго супруга или супруги.

Ваша задача – восстановить по данной информации события, произошедшие в ЗАГСе города N за сегодня.

Формат входных данных

В первой строке дано число n ($n \leqslant 10^5$) – количество записей в журнале.

В каждой из следующих n строк находится либо слово $next$, обозначающее вызов следующей пары, либо фамилия, состоящая из латинских букв и длиной не менее одного и не более двадцати символов.

Каждая из фамилий встречается в записях журнала не менее одного и не более двух раз.

Формат выходных данных

В случае прихода человека, супруг или супруга которого ещё не пришёл или не пришла в ЗАГС, выведите перед фамилией этого человека $1st$. Таким образом мы опишем событие становления в конец очереди нового пришедшего.

В случае прихода человека, супруг или супруга которого уже в ЗАГСе, выведите перед фамилией этого человека $2nd$ без кавычек. Таким образом мы опишем событие появления в очереди пары на месте, которое занял пришедший ранее супруг или супруга.

В случае объявления о вызове, выведите фамилию будущих супругов, чьё заявление будет утверждаться следующим. В случае, если на момент вызова в очереди нет ни одной пары будущих супругов, выводить фамилию будущих супругов, чьё заявление будет утверждаться следующим, следует только после появления этой пары в очереди. Если же до конца дня в ЗАГС не придёт ни одной пары, ничего выводить не нужно.

Не выводите лишние пробелы в конце или начале строк – это будет считаться за ошибку.

Для лучшего понимания формата выходных данных ознакомьтесь с примерами ниже.

Примеры

Пример №1

Стандартный ввод
5 Pit Wait Pit Wait next

Стандартный вывод
1st Pit 1st Wait 2nd Pit 2nd Wait Pit

Пример №2

Стандартный ввод
5 Pit Wait Wait Pit next

Стандартный вывод
1st Pit 1st Wait 2nd Wait 2nd Pit Pit

Пример №3

Стандартный ввод
5
next
Pit
Wait
Wait
Pit

Стандартный вывод
1st Pit
1st Wait
2nd Wait
Wait
2nd Pit

Решение

Разобьём задачу на подзадачи. Что нам нужно уметь делать?

- Проверять, есть ли в очереди человек с определённой фамилией. И если есть, то помечать этого человека.
- Добавлять человека в конец очереди.
- Находить в очереди первого помеченного человека и удалять его из очереди.

Смоделируем очередь с помощью структуры данных очередь (queue). Добавлять в конец очереди новый элемент можно за $O(1)$, но находить помеченный или соответствующий определённой строке-фамилии элемент можно только за $O(n)$, что слишком медленно.

Чтобы быстро (за $O(\log(n))$ или быстрее) находить соответствующий определённой строке-фамилии элемент, будем хранить данные о находящихся сейчас в очереди элементах в словаре (map). Ключами в таком словаре будут строки-фамилии, а значение по определённой строке-фамилии – это положение элемента, соответствующего данной строке-фамилии, в очереди.

Как быстро находить первый помеченный элемент? Вместо того, чтобы в одной очереди хранить и помеченные, и непомеченные элементы, будем в одной очереди хранить помеченные, а в другой – непомеченные элементы. В изначальной очереди будем хранить непомеченные элементы, а вместо того, чтобы помечать некоторый элемент в изначальной очереди, будем переносить этот элемент во вторую очередь.

Но когда нужно будет найти первый помеченный элемент во второй очереди, мы уже не сможем просто взять первый элемент второй очереди. Нам нужно будет найти во второй очереди элемент, который был добавлен в изначальную очередь раньше всего – именно такой элемент должен считаться первым во второй очереди. Для быстрого нахождения и удаления такого элемента заменим структуру данных очередь (queue) на структуру данных очередь с приоритетом (priority queue), в которой в качестве приоритета в пару к элементу поставим целое число, которое будет соответствовать порядковому номеру строки, в которой во входных данных впервые появилась строка-фамилия, которой соответствует элемент. Напомним, что добавление в очередь с приоритетом, как и удаление первого элемента из неё, работает за $O(\log(n))$).

Также отметим, что для отложенных операций удаления первого помеченного элемента из второй очереди нужно поддерживать переменную-счётчик таких отложенных операций и, в случае появления нового элемента во второй очереди, совершать отложенную операцию, если хотя бы одна такая есть.

Таким образом, каждую из операций мы сможем сделать не более чем за $O(\log(n))$ и итоговая асимптотика работы программы будет $O(n \cdot \log(n))$.

Разумеется, существуют и другие способы решения данной задачи.

Пример программы-решения

Ниже представлено решение на языке Java

```java
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.PriorityQueue;
import java.util.StringTokenizer;
import java.util.TreeMap;

public class Main {
    FastScanner in;
    PrintWriter out;

    private void solve() throws IOException {
        int n = in.nextInt();

        String[] q = new String[n];
        int qs = 0, wait = 0;
        TreeMap<String, Integer> map = new TreeMap<>();
        PriorityQueue<Integer> pq = new PriorityQueue<>();

        for (int i = 0; i < n; i++) {
            String s = in.next();
            if (s.equals("next")) {
                if (pq.isEmpty())
                    wait++;
                else
                    out.println(q[pq.remove()]);
            } else {
                if (map.containsKey(s)) {
                    out.println("2nd " + s);
                    pq.add(map.get(s));
                    map.remove(s);
                    if (wait > 0) {
                        out.println(q[pq.remove()]);
                        wait--;
                    }
                } else {
                    out.println("1st " + s);
                    map.put(s, qs);
                    q[qs++] = s;
                }
            }
        }
    }

```

```
    class FastScanner {
        StringTokenizer st;
        BufferedReader br;

        FastScanner(InputStream s) {
            br = new BufferedReader(new InputStreamReader(s));
        }

        String next() throws IOException {
            while (st == null || !st.hasMoreTokens())
                st = new StringTokenizer(br.readLine());
            return st.nextToken();
        }

        boolean hasNext() throws IOException {
            return br.ready() || (st != null && st.hasMoreTokens());
        }

        int nextInt() throws IOException {
            return Integer.parseInt(next());
        }

        long nextLong() throws IOException {
            return Long.parseLong(next());
        }

        double nextDouble() throws IOException {
            return Double.parseDouble(next());
        }

        String nextLine() throws IOException {
            return br.readLine();
        }

        boolean hasNextLine() throws IOException {
            return br.ready();
        }
    }

    private void run() throws IOException {
        in = new FastScanner(System.in);
        out = new PrintWriter(System.out);

        solve();

        out.flush();
        out.close();
    }

    public static void main(String[] args) throws IOException {
        new Main().run();
    }
}
```