

Информатика. Дальний восток. 11.06. 9:30.

Содержание

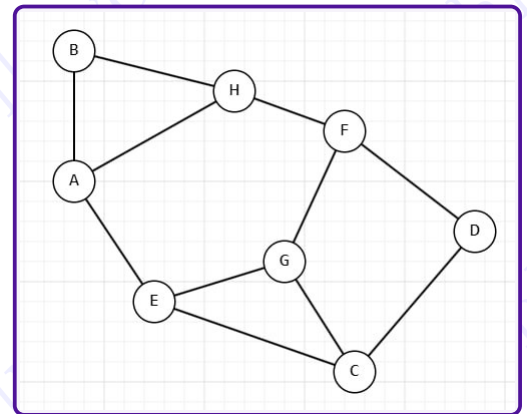
1	Задача 1 (совпадает с ЕГЭ 10.06)	3
2	Задача 2 (совпадает с ЕГЭ 10.06)	4
3	Задача 3.1	5
4	Задача 3.2	6
5	Задача 4	7
6	Задача 5	8
7	Задача 6	9
8	Задача 7	11
9	Задача 8 (совпадает с ЕГЭ 10.06)	11
10	Задача 9	12
11	Задача 10	14
12	Задача 11	14
13	Задача 12.1	15
14	Задача 12.2	17
15	Задача 13	18
16	Задача 14.1	20

17	Задача 14.2	20
18	Задача 15	21
19	Задача 16	22
20	Задача 17	23
21	Задача 18 (совпадает с ЕГЭ 10.06)	24
22	Задача 19 (совпадает с ЕГЭ 10.06)	27
23	Задача 20 (совпадает с ЕГЭ 10.06)	28
24	Задача 21 (совпадает с ЕГЭ 10.06)	29
25	Задача 22 (совпадает с ЕГЭ 10.06)	30
26	Задача 23 (совпадает с ЕГЭ 10.06)	30
27	Задача 24	31
28	Задача 25.1	32
29	Задача 25.2	33
30	Задача 25.3	34
31	Задача 26.1 (совпадает с ЕГЭ 10.06)	35
32	Задача 26.2 (совпадает с ЕГЭ 10.06)	37
33	Задача 26.3 (совпадает с ЕГЭ 10.06)	42
34	Задача 27.1	45
35	Задача 27.2	49

1 Задача 1 (совпадает с ЕГЭ 10.06)

На рисунке схема дорог М-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах). Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова сумма протяжённостей дорог из пункта В в пункт Н и из пункта А в пункт Е. В ответе запишите целое число.

	П1	П2	П3	П4	П5	П6	П7	П8
П1		53		1			2	
П2	53			13				8
П3					30		39	74
П4	1	13				5		
П5			30					2
П6				5			21	
П7	2		39			21		
П8		8	74		3			



Решение

Нам нужно найти уникальную вершину второй степени, которая ведёт в две вершины степени 3, соединяющиеся между собой. Такой уникальной вершиной является В. Значит, вершина В – это П5. Из вершины В ведут две дороги – П3 и П8. Выясним, какая из этих вершин ведёт в вершину степени 3, идущую в вершину степени 2. Вершина П3 идёт в вершину П7, идущую в вершину П6 второй степени. Таким образом: Н – П3, А – П8, F – П7, D – П6, E – П2, G – П4, C – П1. Значит, путь из А в Е равен 8, а путь из В в Н равен 30. Их сумма равна 38.

Ответ: 38

2 Задача 2 (совпадает с ЕГЭ 10.06)

Максим заполнял таблицу истинности логической функции

$$F = z \vee (z \equiv w) \vee \neg(y \rightarrow x)$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
		0	1	0
	1		0	0
	0			0

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z . В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Решение

```
for x in range(2):
    for y in range(2):
        for z in range(2):
            for w in range(2):
                if (z or (z == w) or (not(y <= x))) == False:
                    print(x, y, z, w)
```

Получаем:

```
x y z w
0 0 0 1
1 0 0 1
1 1 0 1
```

В исходной таблице первый столбец полностью свободен – значит, он соответствует столбцу w . Аналогично, только в третьем столбце исходной таблицы нет единиц – это столбец z .

Таблица сейчас выглядит так:

w	?	z	?
1	?	0	1
1	1	0	0
1	0	0	?

Смотрим по строкам: вторая строка соответствует второй из вывода, первая строка может соответствовать только третьей из вывода, тогда третья только первой из вывода. В выводе у переменной x один 0, две 1, а у переменной y два 0, одна 1 \rightarrow второму столбцу соответствует переменная x , четвертому переменная y .

Ответ: $wxyz$

3 Задача 3.1

Используя информацию из приведённой базы данных, определите общую массу (в кг) всех видов мармелада, полученных магазинами, расположенных на улице Заводская, за период с 4 по 19 июня включительно.

В ответе запишите только число.

Решение

На листе Товар фильтруем, оставляя только Мармелад, запоминаем полученные артикулы. На листе Магазин фильтруем, оставляя только магазины на ул. Заводская, запоминаем полученные ID.

Переходим на лист Движение товаров, в столбце Тип операции оставляем только Поступление, в столбце Дата оставляем только период с 4 по 19 июня включительно, в столбце Артикул оставляем выписанные ранее артикулы, в столбце ID магазина оставляем только запомненные ID.

Выделяем кол-во упаковок, запоминаем. Переходим на лист Товар, смотрим, сколько килограммов в одной упаковке — умножаем кол-во упаковок на кол-во килограммов в одной упаковке.

4 Задача 3.2

Используя информацию из приведённой базы данных, определите общую стоимость (в руб.) всех видов карамели, проданных в магазинах, расположенных на улице Заводская, за период с 9 по 18 июня включительно.

В ответе запишите только число.

Решение

На листе Товар фильтруем, оставляя только карамель, запоминаем полученные артикулы. На листе Магазин фильтруем, оставляя только магазины на ул. Заводская, запоминаем полученные ID.

Переходим на лист Движение товаров, в столбце Тип операции оставляем только "Продажа" в столбце Дата оставляем только период с 9 по 18 июня включительно, в столбце Артикул оставляем выписанные ранее артикулы, в столбце ID магазина оставляем только запомненные ID.

Выделяем кол-во упаковок, запоминаем. Переходим на лист Товар, смотрим, сколько стоит одна упаковка в столбце Цена за упаковку — умножаем кол-во упаковок на цену одной упаковки.

5 Задача 4

По каналу связи передаются зашифрованные сообщения, содержащие только семь букв: А, Б, В, Г, Д, Е и Ж. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны:

Буква	Кодовое слово
Г	100
Д	01
Е	000
Ж	001

Какое наименьшее количество двоичных знаков потребуется для кодирования трёх оставшихся букв?

В ответе запишите суммарную длину кодовых слов для букв: А, Б, В.

Примечание. Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Решение

Код не может начинаться с 0, так как ветки 00 и 01 заняты, рассмотрим только коды, начинающиеся с 1. От 10 идут коды 100 и 101, причём 100 занято буквой Г. Запомним, что 101 можно занять. Теперь рассмотрим ветку 11, продлеваем, получаем 110 и 111. Тогда возьмём 101, 110 и 111. Суммарная длина 9, она и будет наименьшей.

Ответ: 9

6 Задача 5

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится троичная запись числа N .
2. Далее эта запись обрабатывается по следующему правилу:
 - а) если число N делится на 3, то слева дописывается «1», а справа — «02»;
 - б) если число N на 3 не делится, то остаток от деления числа на 3 умножается на 4, переводится в троичную запись и дописывается в начало числа.

Полученная таким образом запись является троичной записью искомого числа R .

3. Результат переводится в десятичную систему и выводится на экран.

Например, для исходного числа $11 = 102_3$ результатом является число $10222_3 = 107$, а для исходного числа $12 = 110_3$ результатом является число $11100_3 = 353$.

Укажите минимальное число N , после обработки которого с помощью этого алгоритма получается число R , большее чем 135.

Решение

```
def f(x): # функция для перевода в троичную сс
    s = ''
    while x > 0:
        s += str(x % 3)
        x //= 3
    return s[::-1]
for n in range(1, 1000):
    n_3 = f(n)
    if n % 3 == 0:
        n_3 = '1' + n_3 + '02'
    else:
        n_3 = f((n % 3) * 4) + n_3
    if int(n_3, 3) > 135:
        print(n)
        break
```

Ответ: 6

7 Задача 6

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует 5 команд: **Поднять хвост**, означающая переход к перемещению без рисования; **Опустить хвост**, означающая переход в режим рисования; **Вперёд n** (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; **Назад n** (где n – целое число), вызывающая передвижение в противоположном голове направлении; **Направо t** (где t – целое число), вызывающая изменение направления движения на t градусов по часовой стрелке, **Налево t** (где t – целое число), вызывающая изменение направления движения на t градусов против часовой стрелки. Запись **Повтори k [Команда1 Команда2 ... Команда S]** означает, что последовательность из S команд повторится k раз.

Черепахе был дан для исполнения следующий алгоритм.

Повтори 2 [Вперёд 74 Направо 90 Вперёд 92 Направо 90]

Поднять хвост

Вперёд 3 Направо 90 Вперёд 7 Налево 90

Опустить хвост

Повтори 2 [Вперёд 14 Направо 90 Вперёд 18 Направо 90]

Определите площадь объединения фигур, ограниченного заданными алгоритмом линиями.

Решение

```
from turtle import *
m = 20 # Масштаб
tracer(0)
screensize(5000, 5000) # увеличим размер окна
pd() # опускаем хвост
left(90) # поворачиваем голову в сторону
# положительного направления оси ординат
# Алгоритм
for i in range(2):
    forward(14 * m)
    right(90)
    forward(18 * m)
    right(90)
pu() # поднимаем хвост
forward(3 * m)
right(90)
forward(7 * m)
left(90)
pd() # опускаем хвост
for i in range(2):
    forward(74 * m)
    right(90)
    forward(92 * m)
    right(90)
pu() # поднимаем хвост
# Проставление точек
for x in range(-10, 100):
    for y in range(-10, 110):
        goto(x * m, y * m)
        dot(3)
done()
```

Ответ: 6939

8 Задача 7

Маша делает цветные фотографии на телефон, который сохраняет снимки с размером 3840×2160 пикселей и разрешением 17 бит. После сохранения снимков в памяти телефона Маша отправляет фотографию через мессенджер, который сжимает снимок до размера 1280×720 пикселей, каждый разрешением 5 бит. Какое количество Кбайт удастся сэкономить при отправке 120 фотографий? В ответе запишите целое число.

Решение

1) Найдём объём изображения до сжатия. $V = 3840 \cdot 2160 \cdot 17$. Переведём сразу в Кбайты, получаем $\frac{3840 \cdot 2160 \cdot 17}{2^{13}} = 17212,5$ Кбайта (важно не отбрасывать дробную часть).

2) Найдём объём изображения после сжатия. $V = 1280 \cdot 720 \cdot 5$. В Кбайтах получим $\frac{1280 \cdot 720 \cdot 5}{2^{13}} = 562,5$ Кбайта.

3) Изображение уменьшилось с 17212,5 Кбайта до 562,5 Кбайта. Следовательно, на сжатии одного изображения экономится $17212,5 - 562,5 = 16650$ Кбайт

При отправке 120 фотографий сэкономятся $120 \cdot 16650 = 1998000$ Кбайт

Ответ: 1998000

9 Задача 8 (совпадает с ЕГЭ 10.06)

Все шестибуквенные слова, составленные из букв Т, Е, О, Р, И, Я, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. ЕЕЕЕЕЕ
2. ЕЕЕЕЕИ
3. ЕЕЕЕЕО
4. ЕЕЕЕЕР
5. ЕЕЕЕЕТ

Определите, под каким номером в этом списке стоит последнее слово с нечётным номером, которое не начинается с букв Е, И или О и при этом содержит в своей записи ровно одну букву Т.

Решение

```
from itertools import product
cnt = 1, ans = -1
for word in product('ЕИОРТЯ', repeat=6):
    s = ''.join(word)
    if cnt % 2 != 0 and s[0] not in 'ЕИО' and s.count('Т') == 1:
        ans = cnt
    cnt += 1
print(ans)
```

Ответ: 46655

10 Задача 9

Откройте файл электронной таблицы, содержащей в каждой строке шесть натуральных чисел. Определите наибольший номер строки таблицы, для которой выполнены оба условия:

- в строке есть одно число, которое повторяется трижды, остальные три числа различны;
- повторяющееся число строки больше, чем среднее арифметическое ее неповторяющихся чисел.

В ответе запишите только число.

Решение программой

```
f = open('9_1.txt')
count = 0
for i in f:
    count += 1
    temp = list(map(int, i.strip().split()))
    c3 = [x for x in temp if temp.count(x) == 3]
    c1 = [x for x in temp if temp.count(x) == 1]
    if len(c3) == 3 and len(c1) == 3 and c3[0] > sum(c1) / len(c1):
        print(count)
```

Решение таблицей

Для начала посчитаем сколько раз каждое число встречается в строке. Для этого в ячейке G1 запишем формулу:

$$=СЧЁТЕСЛИ(\$A1:\$F1;A1)$$

и протянем эту формулу до ячейки L1. Теперь проверим, что только одно число встречается трижды в строке и все остальные различные. Для этого в ячейке M1 запишем формулу

$$=ЕСЛИ(И(СЧЁТЕСЛИ(G1:L1;3)=3;СЧЁТЕСЛИ(G1:L1;1)=3);1;0)$$

Теперь посчитаем среднее арифметическое чисел в строке и для этого запишем следующую формулу:

$$=ЕСЛИ(СЧЁТЕСЛИ(G1:L1;1)>0;СРЗНАЧЕСЛИ(G1:L1;1;A1:F1);1000000000)$$

Вычислим число, которое встречается трижды в строке с помощью формулы

$$=СУММЕСЛИ(G1:L1;3;A1:F1)/3$$

Теперь проверим, что число, которое повторяется трижды больше среднего арифметического неповторяющихся чисел с помощью формулы

$$=ЕСЛИ(O1>N1;1;0)$$

Проверим, что оба условия выполняются в строке

$$=ЕСЛИ(И(P1=1;M1=1);1;0)$$

В ячейках R1 и R2 запишем значения 1 и 2 и до конца таблицы продолжим арифметическую прогрессию, теперь мы знаем, в каких строчках выполнялись оба условия. Листаем вниз и видим, что смотрим на какой строке была последняя 1.

11 Задача 10

С помощью текстового редактора определите, сколько раз встречается сочетание букв «вот» или «Вот» только в составе других слов, в том числе в сложных словах, соединённых дефисом, но не как отдельное слово, в тексте повести А.И. Куприна «Гранатовый браслет». В ответе укажите только число.

Решение

Заменяем дефис на любую БУКВУ (главная → заменить → заменить все).

С помощью поиска ищем слово «вот», причём в параметры поиска необходимо указать – «только слова целиком». Таких слов нашлось x .

Теперь снимаем галочку с пункта «только слова целиком» и также ищем слово «вот», не учитывая регистр. Найденных слов – y .

Так как в условии сказано, что отдельные слова не учитываются, то ответ равен $y - x$.

12 Задача 11

На предприятии каждой изготовленной детали присваивают серийный номер, содержащий десятичные цифры, 26 латинских букв (без учёта регистра) и символы из 8164-символьного специального алфавита. В базе данных для хранения каждого серийного номера отведено одинаковое и минимально возможное число байт. При этом используется посимвольное кодирование серийных номеров, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 835 серийных номеров отведено более 156 Кбайт памяти. Определите минимально возможную длину серийного номера. В ответе запишите только целое число.

Решение

Для начала нам нужно определить количество бит, которое необходимо для кодирования алфавита. Общее количество символов в алфавите равняется:

$$10 + 26 + 8164 = 8200 \text{ символов}$$

Для кодирования всех символов понадобится 14 бит.

Теперь определим сколько байт весит один серийный номер, для этого КБайт переведем в байты и поделим на количество серийных номеров:

$$\frac{156 \cdot 2^{10}}{835} = 191,31... \text{ байт.}$$

Округляем в большую сторону, чтобы объем общего количества серийных номеров превышал 156 КБайт памяти и получаем 192 байт.

Вычислим минимальное количество символов в серийном номере:

$$\frac{192 \cdot 8}{14} = 109,71..$$

Округляем в большую сторону, так как если мы округлим в меньшую сторону то общий объем всех серийных номеров не будет превышать 156 КБайт памяти.

Ответ: 110

13 Задача 12.1

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) **заменить** (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды **заменить** (111, 27) преобразует строку 05111150 в строку 0527150. Если в строке нет вхождений цепочки v , то выполнение команды **заменить** (v, w) не меняет эту строку.

Б) **нашлось** (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Дана программа для Редактора:

НАЧАЛО

ПОКА нашлось (055) ИЛИ нашлось (65) ИЛИ нашлось (5555)

ЕСЛИ нашлось (055)

ТО заменить (055, 56)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (65)

ТО заменить (65, 5)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (5555)

ТО заменить (5555, 000)

КОНЕЦ ЕСЛИ

ЕСЛИ нашлось (00000)

ТО заменить (00000, 6)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, содержащая n цифр «5» и заканчивающаяся цифрой «0» ($3 < n < 2500$).

Определите наибольшее возможное значение суммы числовых значений цифр в строке, которая может быть результатом выполнения данной программы.

Решение

```
a = set() # Создаём множество для сумм цифр
for n in range(4, 2500): # Перебираем количество символов '5'
    s = n * '5' + '0' # Получаем строку
    while ('055' in s) or ('65' in s) or ('5555' in s):
        if '055' in s:
            s = s.replace('055', '56', 1)
        if '65' in s:
            s = s.replace('65', '5', 1)
        if '5555' in s:
            s = s.replace('5555', '000', 1)
        if '00000' in s:
            s = s.replace('00000', '6', 1)
    a.add(sum(map(int, s))) # В множество добавляем сумму цифр строки
print(max(a)) # Выводим максимальную сумму цифр
```

Ответ: 15

14 Задача 12.2

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнить две команды, в обеих командах v и w обозначают цепочки цифр:

А) заменить (v, w) — эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды заменить (111, 27) преобразует строку 05111150 в строку 0527150. Если в строке нет вхождения цепочки v , то выполнение команды **заменить** (v, w) не меняет эту строку.

Б) нашлось (v) — эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Данные программы для Редактора:

```
ПОКА нашлось (12) ИЛИ нашлось (322) ИЛИ нашлось (222)
```

```
  ЕСЛИ нашлось (12)
```

```
    ТО заменить (12, 2)
```

```
  КОНЕЦ ЕСЛИ
```

```
  ЕСЛИ нашлось (322)
```

```
    ТО заменить (322, 21)
```

```
  КОНЕЦ ЕСЛИ
```

```
  ЕСЛИ нашлось (222)
```

```
    ТО заменить (222, 3)
```

```
  КОНЕЦ ЕСЛИ
```

```
КОНЕЦ ПОКА
```

На вход приведенной выше программе поступает строка, начинающаяся с цифры «1», а затем содержащая n цифр «2» ($3 < n < 10000$).

Определите значение n , при котором сумма числовых значений цифр в строке, являющейся результатом работы программы, максимальна.

Решение

```
max_sum = 0
max_n = 0
for n in range(4, 10_000): # Перебираем количество символов '2'
    s = '1' + '2' * n # Получаем строку
    # Осуществляем алгоритм
    while ('12' in s) or ('322' in s) or ('222' in s):
        if '12' in s:
            s = s.replace('12', '2', 1)
        if '322' in s:
            s = s.replace('322', '21', 1)
        if '222' in s:
            s = s.replace('222', '3', 1)
    sum_digits = sum(map(int, s)) # Находим сумму цифр строки
    if sum_digits > max_sum:
        max_sum = sum_digits
        max_n = n
print(max_n) # Выводим n, при котором макс. сумма цифр
```

Ответ: 50

15 Задача 13

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая - к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и его маске. Широковещательным адресом называется специализированный адрес, в котором на месте нулей в маске стоят единицы. Адрес сети и широковещательный адрес не могут быть использованы для адресации сетевых устройств. Сеть задана IP-адресом одного из входящих в неё узлов 143.168.72.213 и сетевой маской 255.255.255.240.

Определите наибольший IP-адрес данной сети, который может быть присвоен компьютеру. В ответе укажите найденный IP-адрес без разделителей.

Решение программой

```
from ipaddress import *
```

```
net = ip_network('143.168.72.213/255.255.255.240', 0)
```

```
print(net[-2])
```

```
"""
```

```
net - объект сети, содержит все её адреса по возрастанию;  
последний адрес в сети совпадает с широковещательным,  
а он не может быть присвоен компьютеру, записываем предпоследний  
"""
```

Решение руками

Переведём IP-адрес узла и маску сети в двоичную систему счисления:

IP узла 10001111.10101000.01001000.11010101

Маска 11111111.11111111.11111111.11110000

Узлы в сети имеют вид:

10001111.10101000.01001000.1101xxxx (так как последние 4 цифры маски - нули)

Необходимо найти наибольший IP-адрес в данной сети, который может быть назначен компьютеру. Наибольший ip-адрес получим, подставив все единички в узловой части. Однако тогда найдём широковещательный адрес, который НЕ может быть назначен компьютеру. Возьмём адрес на 1 меньше широковещательного, в узловой части будет 1110. Полный адрес имеет вид:

10001111.10101000.01001000.11011110.

Для ответа переведём каждый байт в десятичную систему счисления и запишем без точек: 14316872222.

Ответ: 14316872222

16 Задача 14.1

Операнды арифметического выражения записаны в системе счисления с основанием 39.

$$653x71_{39} + 42x37_{39}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 39-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения в десятичной системе счисления кратно 14. Для найденного значения x вычислите частное от деления значения арифметического выражения на 14 и укажите его в ответе. Основание системы счисления в ответе указывать не нужно.

Решение

```
for x in range(39):
    s1 = 6*39**5 + 5*39**4 + 3*39**3 + x*39**2 + 7*39**1 + 1*39**0
    s2 = 4*39**4 + 2*39**3 + x*39**2 + 3*39**1 + 7*39**0
    if (s1+s2) % 14 == 0:
        print((s1+s2)//14)
```

Ответ: 40176157

17 Задача 14.2

Операнды арифметического выражения записаны в системе счисления с основанием 37.

$$32x437_{37} + 5x2937_{37}$$

В записи чисел переменной x обозначена неизвестная цифра из алфавита 37-ричной системы счисления. Определите наименьшее значение x , при котором значение данного арифметического выражения в десятичной системе счисления кратно 63. Для найденного значения x вычислите частное от деления значения арифметического выражения на 63 и укажите его в ответе. Основание системы счисления в ответе указывать не нужно.

Решение

```
for x in range(37):
    s1 = 3 * 37**5 + 2 * 37**4 + x * 37**3 + 4 * 37**2 + 3 * 37**1 + 7
    s2 = 5 * 37**5 + x * 37**4 + 2 * 37**3 + 9 * 37**2 + 3 * 37**1 + 7
    if (s1 + s2) % 63 == 0:
        print(x, (s1 + s2) // 63)
```

Ответ: 9966867

18 Задача 15

Точного условия нет, похоже на задание майской апробации:

Для какого наименьшего целого неотрицательного числа A логическое выражение

$$(x \geq 9) \vee (2x < y) \vee (xy < A)$$

тождественно истинно (т.е. принимает значение 1) при любых целых неотрицательных x и y ?

Решение

```
for A in range(1000):
    flag = 0
    for x in range(500):
        for y in range(500):
            f = (x >= 9) or (2 * x < y) or (x*y < A)
            if f == 0:
                flag = 1
                break
        if flag == 1:
            break
    if flag == 0:
        print(A)
        break
```

Ответ: 129

19 Задача 16

Алгоритмы вычисления значения функции $F(n)$ и $G(n)$, где n – целое число, заданы следующими соотношениями:

$$F(n) = n \text{ при } n \leq 7;$$

$$F(n) = G(n - 3) \times 3, \text{ если } n > 7.$$

$$G(n) = n \text{ при } n \leq 7;$$

$$G(n) = G(n - 1) + 4, \text{ если } n > 7.$$

Чему равно значение выражения $F(43000)$?

Решение

```
from functools import lru_cache
@lru_cache(None)
def f(n):
    if n <= 7:
        return n
    return g(n - 3) * 3

@lru_cache(None)
def g(n):
    if n <= 7:
        return n
    return g(n - 1) + 4

for i in range(43000):
    f(i)
    g(i)

print(f(43000))
```

Ответ: 515901

20 Задача 17

В файле содержится последовательность целых чисел. Её элементы могут принимать целые значения от -100 000 до 100 000 включительно. Определите количество троек элементов последовательности, в которых хотя бы два числа являются двузначными, а квадрат суммы чисел тройки не больше максимального элемента последовательности, оканчивающегося на 33. В ответе запишите количество найденных троек, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Решение

```
f = open('...')
a = [int(x) for x in f]
max_33 = -10 ** 10 # для хранения макс. числа, оканчивающегося на 33
for x in a:
    # не забываем модуль для отрицательных чисел
    if abs(x) % 100 == 33:
        max_33 = max(max_33, x)
cnt = 0
max_sum = 0
for i in range(len(a) - 2):
    # проверяем, что хотя бы для 2 чисел условие двузначности выполнится
    if ((10 <= abs(a[i]) <= 99) + (10 <= abs(a[i + 1]) <= 99)
        + (10 <= abs(a[i + 2]) <= 99)) >= 2:
        # проверяем квадрат суммы
        if (a[i] + a[i + 1] + a[i + 2]) ** 2 <= max_33:
            cnt += 1
            max_sum = max(max_sum, a[i] + a[i + 1] + a[i + 2])
print(cnt, max_sum)
```

21 Задача 18 (совпадает с ЕГЭ 10.06)

Исполнитель Робот стоит в левом верхнем углу поля, разлинованного на клетки. Он может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: **вправо** или **вверх**. По команде вправо Робот перемещается в соседнюю правую клетку; по команде вниз – в соседнюю нижнюю. Между соседними клетками квадрата также могут быть внутренние стены. Сквозь стену Робот пройти не может. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клеткам маршрута Робота. В «угловых» клетках поля – тех, которые справа и сверху ограничены стенами, Робот не может продолжать движение, поэтому накопленная сумма считается итоговой. Таких конечных клеток на поле может быть несколько, включая правую верхнюю клетку поля. При разных запусках итоговые накопленные суммы могут различаться. Определите максимальную и минимальную денежные суммы, среди всех возможных итоговых сумм, которые может собрать Робот, пройдя из левой нижней клетки в конечную клетку маршрута.

Исходные данные записаны в файле в виде электронной таблицы, каждая ячейка которой соответствует клетке поля. В ответе запишите два числа – сначала максимальную сумму, которую может собрать Робот, затем – минимальную.

Решение

Переносим форматирование, вставляем исходное значение.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	77	68	28	32	50	17	89	22	48	42	13	43	92	9	95	24	56	30	82	78	
2	12	5	10	77	93	20	35	44	49	28	34	12	8	6	35	98	54	46	87	56	
3	100	90	54	41	96	1	8	81	100	55	13	25	2	37	47	42	48	58	70	51	
4	26	38	29	87	8	93	46	21	47	21	56	6	81	58	79	40	15	16	35	37	
5	93	82	73	13	58	58	29	14	62	73	48	92	49	71	12	49	73	43	52	68	
6	50	20	97	3	51	20	16	57	70	10	18	76	86	41	74	48	11	9	2	85	
7	49	34	97	60	41	93	57	14	70	85	4	56	97	48	39	94	72	94	64	1	
8	84	79	63	10	60	62	5	70	49	65	57	17	13	99	68	64	72	84	36	38	
9	63	6	28	26	44	19	76	88	66	78	1	88	98	68	90	12	22	3	96	27	
10	68	38	44	11	28	100	4	73	27	5	52	7	94	92	39	82	59	94	30	48	
11	74	23	42	7	95	63	43	2	21	20	31	18	54	54	22	26	13	75	5	30	
12	2	97	97	24	11	54	49	78	46	15	18	2	61	33	68	10	42	31	26	27	
13	79	59	43	88	29	56	64	43	89	28	7	16	54	77	81	79	25	36	20	86	
14	87	71	3	60	17	2	9	26	64	3	81	61	38	51	43	47	86	50	49	91	
15	81	37	3	30	80	47	17	2	23	83	15	26	88	93	84	6	45	31	15	53	
16	86	37	7	41	35	68	11	36	58	58	5	90	10	63	30	4	55	78	32	87	
17	91	54	11	98	73	95	1	35	79	68	73	57	67	26	76	20	43	62	49	90	
18	79	21	59	62	57	98	21	68	47	76	66	28	46	3	70	12	2	40	90	56	
19	55	44	59	56	86	62	81	94	17	62	38	86	67	64	1	21	42	86	27	73	
20	43	24	49	89	5	58	88	100	19	82	55	43	33	89	85	23	31	32	12	38	
21																					
22																					
23																					
24																					
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					
34																					
35																					
36																					
37																					
38																					
39																					
40																					
41																					
42																					
43	43																				
44																					

В левый столбец динамически к сумме прибавляем значение из соответствующей ячейки

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	77	68	26	32	50	17	69	22	48	42	13	43	92	9	95	24	56	90	82	78
2	12	5	10	77	93	20	35	44	49	28	34	12	8	6	35	98	54	46	87	56
3	100	90	54	41	96	1	8	81	100	55	13	25	2	37	47	42	48	58	70	51
4	26	38	29	67	6	93	46	21	47	21	56	6	81	58	79	40	15	16	35	87
5	93	82	73	13	58	58	29	14	62	73	48	92	49	71	12	49	73	43	52	68
6	50	20	97	3	51	20	16	57	70	10	18	76	86	41	74	48	12	9	2	85
7	49	34	97	60	41	93	57	14	70	85	4	56	97	48	39	94	72	94	64	1
8	84	79	63	10	60	62	5	70	49	65	57	17	13	99	68	64	72	84	36	38
9	63	6	28	26	44	19	76	88	66	78	1	88	98	68	90	12	22	3	96	27
10	68	38	44	11	28	100	4	73	27	5	52	7	94	92	39	82	59	94	30	48
11	74	23	42	7	95	63	45	2	21	20	31	18	54	54	22	26	19	75	5	30
12	2	97	97	24	11	54	49	78	46	15	18	2	61	33	68	19	42	31	26	27
13	79	59	43	88	29	56	64	43	89	28	7	16	54	77	81	79	25	36	20	86
14	87	71	3	60	17	2	9	26	64	3	81	61	38	51	43	47	86	50	49	91
15	81	37	3	30	80	47	17	2	23	83	15	26	88	93	84	6	45	31	15	53
16	86	37	7	41	35	68	11	36	58	58	5	90	10	63	30	4	55	78	32	87
17	91	54	11	98	73	95	1	75	79	68	73	57	67	24	76	20	43	62	49	90
18	79	21	59	62	57	98	21	68	47	76	66	28	46	3	70	12	2	40	90	56
19	55	44	59	56	66	62	81	94	17	62	38	86	67	64	1	21	42	86	27	73
20	43	24	49	89	5	58	88	100	19	82	55	43	33	89	85	23	31	32	12	38
21																				
22																				
23	1299																			
24	1222																			
25	1210																			
26	1110																			
27	1084																			
28	991																			
29	941																			
30	892																			
31	808																			
32	745																			
33	677																			
34	603																			
35	601																			
36	522																			
37	435																			
38	354																			
39	268																			
40	177																			
41	=A42+A19																			
42	43																			
43																				
44																				

Для подсчета максимума берем максимум из двух сумм, прибавляем значение из соответствующей ячейки, растягиваем на всю таблицу.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T			
1	77	68	26	32	50	17	69	22	48	42	13	43	92	9	95	24	56	90	82	78			
2	12	5	10	77	93	20	35	44	49	28	34	12	8	6	35	98	54	46	87	56			
3	100	90	54	41	96	1	8	81	100	55	13	25	2	37	47	42	48	58	70	51			
4	26	38	29	67	6	93	46	21	47	21	56	6	81	58	79	40	15	16	35	87			
5	93	82	73	13	58	58	29	14	62	73	48	92	49	71	12	49	73	43	52	68			
6	50	20	97	3	51	20	16	57	70	10	18	76	86	41	74	48	12	9	2	85			
7	49	34	97	60	41	93	57	14	70	85	4	56	97	48	39	94	72	94	64	1			
8	84	79	63	10	60	62	5	70	49	65	57	17	13	99	68	64	72	84	36	38			
9	63	6	28	26	44	19	76	88	66	78	1	88	98	68	90	12	22	3	96	27			
10	68	38	44	11	28	100	4	73	27	5	52	7	94	92	39	82	59	94	30	48			
11	74	23	42	7	95	63	45	2	21	20	31	18	54	54	22	26	19	75	5	30			
12	2	97	97	24	11	54	49	78	46	15	18	2	61	33	68	19	42	31	26	27			
13	79	59	43	88	29	56	64	43	89	28	7	16	54	77	81	79	25	36	20	86			
14	87	71	3	60	17	2	9	26	64	3	81	61	38	51	43	47	86	50	49	91			
15	81	37	3	30	80	47	17	2	23	83	15	26	88	93	84	6	45	31	15	53			
16	86	37	7	41	35	68	11	36	58	58	5	90	10	63	30	4	55	78	32	87			
17	91	54	11	98	73	95	1	75	79	68	73	57	67	24	76	20	43	62	49	90			
18	79	21	59	62	57	98	21	68	47	76	66	28	46	3	70	12	2	40	90	56			
19	55	44	59	56	66	62	81	94	17	62	38	86	67	64	1	21	42	86	27	73			
20	43	24	49	89	5	58	88	100	19	82	55	43	33	89	85	23	31	32	12	38			
21																							
22																							
23	1299																						
24	1222																						
25	1210																						
26	1110																						
27	1084																						
28	991																						
29	941																						
30	892																						
31	808																						
32	745																						
33	677																						
34	603																						
35	601																						
36	522																						
37	435																						
38	354																						
39	268																						
40	177																						
41	=МАКС(A41; B42)+B19																						
42	43	67			116	205	210	268	356	456	475	557	612	655	688	777	862	885	916	948	960	998	
43																							
44																							

22 Задача 19 (совпадает с ЕГЭ 10.06)

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может:

- убрать из кучи 3 камня;
- убрать из кучи 6 камней;
- уменьшить количество камней в куче в 3 раза (количество камней, полученное при делении, округляется до меньшего).

Например, из кучи в 20 камней за один ход можно получить кучу из 17, 14 или 6 камней.

Игра завершается, когда количество камней в куче становится не более 25. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу из 25 или меньше камней. В начальный момент в куче было S камней, $S \geq 26$.

Укажите минимальное значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. может выиграть своим первым ходом.

Решение

```
from functools import lru_cache
@lru_cache(None)
def game(x):
    if x <= 25: # условие выигрыша
        return 0
    moves = [game(x - 3), game(x - 6)] # перечисляем ходы
    if x > 0: moves.append(game(x // 3))
    n = [i for i in moves if i <= 0]
    if n: return -max(n) + 1
    return -max(moves)
for i in range(26, 100):
    if game(i) == -1:
        print(i)
```

Ответ: 78

23 Задача 20 (совпадает с ЕГЭ 10.06)

Для игры, описанной в задании 19, найдите два наименьших значения S , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

Решение

```
from functools import lru_cache
@lru_cache(None)

def game(x):
    if x <= 25: # условие выигрыша
        return 0
    moves = [game(x - 3), game(x - 6)] # перечисляем ходы
    if x > 0:
        moves.append(game(x // 3))
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)

for i in range(26, 100):
    if game(i) == 2:
        print(i)
```

Ответ: 81 82

24 Задача 21 (совпадает с ЕГЭ 10.06)

Для игры, описанной в задании 19, найдите минимальное значение S , при котором одновременно выполняются условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Решение

```
from functools import lru_cache
@lru_cache(None)
def game(x):
    if x <= 25: # условие выигрыша
        return 0
    moves = [game(x - 3), game(x - 6)] # перечисляем ходы
    if x > 0:
        moves.append(game(x // 3))
    n = [i for i in moves if i <= 0]
    if n:
        return -max(n) + 1
    return -max(moves)
for i in range(26, 100):
    if game(i) == -2:
        print(i)
```

Ответ: 87

25 Задача 22 (совпадает с ЕГЭ 10.06)

В файле 22.xls содержится информация о совокупности 25 вычислительных процессов, которые могут выполняться параллельно или последовательно. Будем говорить, что процесс В зависит от процесса А, если для выполнения процесса В необходимы результаты выполнения процесса А. В этом случае процессы могут выполняться только последовательно. Информация о процессах представлена в файле в виде таблицы. В первом столбце таблицы указан идентификатор процесса (ID), во втором столбце таблицы – время его выполнения в миллисекундах, в третьем столбце перечислены с разделителем «;» ID процессов, от которых зависит данный процесс. Если процесс является независимым, то в таблице указано значение 0.

Определите, сколько процессов выполнялось одновременно в 14 мс, если время начала каждого процесса минимально.

В 1-й день практически у всех был такой прототип, только с разными мс.

Решение

Для независимых процессов, начиная со столбца D, будем проставлять единицы вправо. Количество единиц равно количеству мс выполнения каждого процесса. Для удобства выберем ширину столбцов 2-3.

Затем переходим к зависимым процессам и начинаем также проставлять единицы, но уже после выполнения независимых процессов.

В последней после таблицы строке столбца D прописываем формулу СУММ(D2:D13) и растягиваем её вправо. Полученные значения – количество выполняемых одновременно процессов. Посчитаем сколько процессов выполнялось одновременно в 14 мс.

26 Задача 23 (совпадает с ЕГЭ 10.06)

Исполнитель преобразует число на экране. У исполнителя есть две команды, которые обозначены латинскими буквами:

А. Вычти 3

В. Найди целую часть от деления на 3

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 81 результатом является число 3, и при этом траектория вычислений содержит число 27?

Решение

```
def f(a, b):
    if a < b:
        return 0
    if a == b:
        return 1
    return f(a - 3, b) + f(int(a / 3), b)
print(f(81, 27) * f(27, 3))
```

Ответ: 10

27 Задача 24

Файл состоит из букв латинского алфавита и десятичных цифр. Определите максимальную длину подцепочки, которая содержит комбинацию цифр 2025 (в указанном порядке) ровно 60 раз, при этом искомая последовательность начинается на комбинацию цифр 2025 (в указанном порядке) и содержит не менее 120 букв Y.

Решение

```
s = open("24_2.txt").readline()
st = s[:4] != '2025' # начинается ли файл с "2025"
s = s.split("2025")
mx = 0

for i in range(st, len(s) - 60):
    # отдельно укажем первую "2025", так как
    # join не добавляет в начало, только между элементами
    substr = '2025' + "2025".join(s[i:i + 60])
    # сейчас ровно 60 "2025", проверяем количество Y
    if substr.count("Y") >= 120:
        if len(substr) + 3 > mx:
            # прибавляем 3, так как справа из 2025 можем взять 202
            mx = max(mx, len(substr) + 3)

print(mx)
```

28 Задача 25.1

Пусть M – сумма минимального и максимального простых делителей целого числа, не считая самого числа. Если у числа нет простых делителей, то $M = 0$.

Определите первые пять чисел, большие, чем 7 000 000, у которых оканчивается на 13.

Решение

```
def prime(x):
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            return False
    return True

def divs(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

count = 0
for x in range(7_000_001, 10_000_000):
    d = [i for i in divs(x) if prime(i)]
    if len(d) > 0:
        M = min(d) + max(d)
        if M % 100 == 13:
            print(x)
            count += 1
            if count > 4:
                break
```

29 Задача 25.2

Пусть R — сумма максимального и минимального простых делителей целого числа, не считая единицы и самого числа. Если у числа нет простых делителей, то считаем значение R равным нулю. Напишите программу, которая перебирает целые числа, большие 6000000, в порядке возрастания и ищет среди них первые шесть таких чисел, для которых R заканчивается на 19.

Решение

```
def prime(x):
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            return False
    return x > 1

def divs(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

count = 0
for x in range(6_000_001, 10_000_000):
    d = [i for i in divs(x) if prime(i)]
    if len(d) > 0:
        R = min(d) + max(d)
        if R % 100 == 19:
            print(x)
            count += 1
            if count > 5:
                break
```

30 Задача 25.3

Пусть M — сумма максимального и минимального простых делителей целого числа, не считая единицы и самого числа. Если у числа нет простых делителей, то считаем значение M равным нулю. Напишите программу, которая перебирает целые числа, большие 7 500 000, в порядке возрастания и ищет среди них первые пять таких чисел, для которых M заканчивается на 31 и кратно общему количеству простых делителей.

Решение

```
def prime(x):
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            return False
    return x > 1

def divs(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

count = 0
for x in range(7_500_001, 10_000_000):
    d = [i for i in divs(x) if prime(i)]
    if len(d) > 0:
        M = max(d) + min(d)
        if M % 100 == 31 and M % len(d) == 0:
            print(x)
            count += 1
            if count > 4:
                break
```

31 Задача 26.1 (совпадает с ЕГЭ 10.06)

Логистическая компания использует контейнеры с максимальной вместимостью M кг. Для отправки N грузов контейнеры заполняют грузами в порядке убывания веса, пока общая масса грузов не превысит максимальную вместимость M . Затем процедуру повторяют для следующего контейнера, пока все грузы не будут размещены. Необходимо определить количество контейнеров для отправки всех грузов и общую загрузку предпоследнего контейнера.

Входные данные

Входные данные представлены в файле следующим образом. В первой строке записаны два числа – количество грузов N и максимальная вместимость контейнера M . В каждой из следующих N строк записано по одному натуральному числу, обозначающему вес груза.

Пример входного файла

```
6 100
30
10
40
50
10
20
```

В первый контейнер будут помещены грузы весом 50, 40 и 10 кг (общий вес 100 кг), во второй контейнер – грузы весом 30, 20 и 10 кг (общий вес 60 кг). Ответ: 2 100.

Решение

```
with open("26.txt") as file:
    n, m = map(int, file.readline().split())
    weights = sorted([int(line) for line in file], reverse=True)

container_count = 0
res = []
while weights:
    current_weight = 0
    current_ans = []
    for weight in weights[:]:
        if current_weight + weight <= m:
            current_weight += weight
            weights.remove(weight)
            current_ans.append(weight)
    container_count += 1
    res.append(current_ans)

print(container_count, sum(res[-2]))
```

Ответ: 86 149

32 Задача 26.2 (совпадает с ЕГЭ 10.06)

В магазине для упаковки подарков есть N кубических коробок красного цвета и M кубических коробок зеленого цвета ($N > M$). Самой интересной считается упаковка подарка по принципу матрешки – подарок упаковывается в одну из коробок, та в свою очередь в другую коробку и т. д., при этом цвет коробок чередуется. Одну коробку можно поместить в другую, если длина её стороны хотя бы на 5 единиц меньше длины стороны другой коробки. Определите наибольшее количество коробок, которое можно использовать для упаковки одного подарка, и максимально возможную длину стороны самой маленькой коробки, где будет находиться подарок. Размер подарка позволяет поместить его в самую маленькую коробку.

Входные данные: В первой строке входного файла находятся число N — количество коробок красного цвета в магазине (натуральное число, не превышающее 10 000) и через пробел число M — количество коробок зеленого цвета в магазине (натуральное число, не превышающее 10 000). В следующих N строках находятся значения длин сторон коробок красного цвета (все числа натуральные, не превышающие 10 000) и через знак табуляции значения длин сторон коробок зеленого цвета (все числа натуральные, не превышающие 10 000), каждая пара таких значений — в отдельной строке; в последних $N - M$ строках второе число опускается, и числа, соответствующие длинам сторон коробок красного цвета, идут каждое в отдельной строке.

Запишите в ответе два целых числа: сначала наибольшее количество коробок, которое можно использовать для упаковки одного подарка, затем максимально возможную длину стороны самой маленькой коробки в таком наборе.

Типовой пример организации данных во входном файле

5 4

39 55

40 42

44 44

40 55

50

Пример входного файла приведён для случая пяти коробок красного цвета и четырёх коробок синего цвета, когда минимальная допустимая разница между длинами

сторон коробок, подходящих для упаковки «матрёшкой», составляет 3 единицы. При таких исходных данных ответом будет являться 4 40.

Решение

Таблицами:

Добавляем обозначение цвета, пусть 0 – красный, 1 – синий

	A	B	C	D
1	5		4	
2	39	0	55	1
3	40	0	42	1
4	44	0	44	1
5	40	0	55	1
6	50	0		
7				

Переносим одно под другое, сортируем по убыванию столбец с матрёшками.

	A	B
1	55	1
2	55	1
3	50	0
4	44	0
5	44	1
6	42	1
7	40	0
8	40	0
9	39	0
10		
11		

В первую строку берем самую большую коробку с цветом.

Во вторую строку записываем формулу для размера:

=ЕСЛИ(C1-A2>=5; ЕСЛИ(ИЛИ(И(D1=1; B2=0); И(D1=0; B2=1))); A2; C1); C1)

Что происходит в формуле:

1) C1-A2>=5

Проверяем, достаточно ли велика разница между предыдущей коробкой (C1) и текущей (A2). Если разница >=5, то коробка может поместиться

2) Если разница достаточная, проверяем цвет:

ИЛИ(И(D1=1; B2=0); И(D1=0; B2=1))

И(D1=1; B2=0) - предыдущая синяя (1) И текущая красная (0)

И(D1=0; B2=1) - предыдущая красная (0) И текущая синяя (1)

ИЛИ - любое из этих условий = цвета чередуются

3) Результат:

Если разница >=5 И цвета чередуются -> берем новую коробку A2

Если разница >=5 НО цвета НЕ чередуются -> оставляем старую C1

Если разница <5 -> оставляем старую $C1$

В столбец D записываем формулу для цвета:

$$=ЕСЛИ(C1=C2; D1; B2)$$

$C1=C2$ - проверяем, изменился ли размер коробки

Если НЕ изменился ($C1=C2$) -> цвет остается тот же $D1$

Если изменился (взяли новую коробку) -> цвет новый $B2$

	A	B	C	D	E	F	G	H	I
1	55	1	55	1					
2	55	1	C1	1					
3	50	0	50	0					
4	44	0	50	0					
5	44	1	44	1					
6	42	1	44	1					
7	40	0	44	1					
8	40	0	44	1					
9	39	0	39	0					
10									
11									

Чтобы посчитать кол-во коробок, посмотрим: если размер коробки не меняется – значит, и кол-во коробок не увеличивается. Динамически посчитаем кол-во коробок:

$$=ЕСЛИ(C2=C1; E1; E1+1)$$

	A	B	C	D	E	F
1	55	1	55	1	1	
2	55	1	55	1	1	
3	50	0	50	0	2	
4	44	0	50	0	2	
5	44	1	44	1	3	
6	42	1	44	1	3	
7	40	0	44	1	3	
8	40	0	44	1	3	
9	39	0	39	0	4	
10						

Тогда кол-во будет равно максимальному, последнему значению, для подсчета наименьшей нужно посмотреть на последнюю взятую возможную коробку: на примере на коробки со значениями 4 в столбце E : если бы их было несколько – взяли бы среди них ту, что имеет наибольшее значение.

Порогой:

```
f = open("26.txt")
first_line = f.readline().strip()
n, m = map(int, first_line.split())

red_boxes = []
green_boxes = []

for i in range(m):
    line = f.readline().strip()
    parts = line.split()
    red_boxes.append(int(parts[0]))
    green_boxes.append(int(parts[1]))
for i in range(n - m):
    line = f.readline()
    red_boxes.append(int(line))

all_boxes = []

# красные коробки (цвет = 0)
for size in red_boxes:
    all_boxes.append((size, 0))

# зеленые коробки (цвет = 1)
for size in green_boxes:
    all_boxes.append((size, 1))

# сортируем по убыванию размера
all_boxes.sort(reverse=True)

# находим максимальную матришку
max_boxes = 0
max_min_size = 0
```

```
# пробуем начать с каждой коробки
for start in range(len(all_boxes)):
    # начинаем новую последовательность
    matryoshka = [all_boxes[start]]
    current_size = all_boxes[start][0]
    current_color = all_boxes[start][1]

    # ищем подходящие коробки для продолжения
    for i in range(start + 1, len(all_boxes)):
        next_size, next_color = all_boxes[i]

        # проверяем условия:
        # 1. разница в размерах >= 5
        # 2. цвета чередуются
        if current_size - next_size >= 5 and next_color != current_color:
            matryoshka.append(all_boxes[i])
            current_size = next_size
            current_color = next_color

    # обновляем максимум
    if len(matryoshka) > max_boxes:
        max_boxes = len(matryoshka)
        max_min_size = matryoshka[-1][0]
    elif len(matryoshka) == max_boxes and matryoshka[-1][0] > max_min_size:
        max_min_size = matryoshka[-1][0]

print(max_boxes, max_min_size)
```

33 Задача 26.3 (совпадает с ЕГЭ 10.06)

Входной файл содержит сведения о заявках на проведение занятий в конференц-зале. В каждой заявке указаны время начала и время окончания мероприятия (в минутах от начала суток). Если время начала одного мероприятия меньше времени окончания другого, то провести можно только одно из них. Если время окончания одного мероприятия совпадает с временем начала другого, то провести можно оба. Определите максимальное количество мероприятий, которое можно провести в конференц-зале и самое позднее время окончания последнего мероприятия.

Входные данные представлены в файле следующим образом. Первая строка входного файла содержит натуральное число N ($1 \leq N \leq 1000$) – количество заявок на проведение мероприятий. Следующие N строк содержат пары чисел, обозначающих время начала и время окончания мероприятий. Каждое из чисел натуральное, не превосходящее 1440.

Запишите в ответе два числа: максимальное количество мероприятий, которое можно провести в конференц-зале и самое раннее время начала последнего мероприятия (в минутах от начала суток).

Пример входного файла:

5

10 150

100 110

131 170

131 180

120 130

При таких исходных данных можно провести максимум три мероприятия, например, по заявкам 2, 3 и 5. Последнее мероприятие в конференц-зала начнётся самое раннее на 131-й минуте, если состоятся мероприятия по заявкам 2, 4, 5. Ответ: 3 131.

Решение

Решение при помощи электронных таблиц:

Сначала переносим данные в Excel, удалим первую строчку, а потом при помощи настраиваемой сортировки отсортируем в первую очередь по столбцу В, а потом по столбцу А - всё по возрастанию. Нужно сортировать по времени окончания мероприятия, так как чем раньше одно кончится, тем раньше сможет начаться следующее.

В ячейку С1 запишем значение ячейки В1, в ячейку С2 запишем следующую формулу и растянем её до конца таблицы:

$$=ЕСЛИ(А2>=С1;В2;С1)$$

Таким образом у нас в столбце С будет храниться время окончания последнего (на данный момент) мероприятия. Каждое новое мероприятие мы проверяем - может ли оно начаться после предыдущего последнего, и если да - сохраняем уже новое время окончания, иначе дублируем старое.

В столбце D мы посчитаем количество прошедших мероприятий. В ячейку D1 запишем 1, а в ячейку D2 запишем следующую формулу и растянем её до конца таблицы:

$$=ЕСЛИ(С2<>С1;D1+1;D1)$$

Если сменилось время окончания – увеличиваем счётчик, иначе дублируем предыдущий.

Максимальное число в столбце D и будет количеством мероприятий.

Минимальное допустимое время начала последнего мероприятия найдём при помощи следующей формулы, в качестве диапазона выбрав отрезок столбца А напротив максимального числа в столбце С:

$$=МИНЕСЛИ(А903:А966;А903:А966;">="&С902)$$

Не обязательно, что число именно такое, приведено в качестве примера.

Решение при помощи программы:

```
f = open('3_26_conf.txt')
n = int(f.readline())

confs = [list(map(int, i.split())) for i in f]

# Сортируем по времени окончания, так как
# чем раньше одно кончится, тем раньше сможет начаться следующее
confs.sort(key=lambda x: (x[1], x[0]))

# Время окончания предпоследнего мероприятия
prev_end = 0
# Время окончания последнего мероприятия
last_end = confs[0][1]
# Все возможные времена начала последнего мероприятия
last_starts = [confs[0][0]]
cnt = 1

for start, end in confs:
    # Может начаться новое - начинаем, не может - добавляем в список возможных
    if start >= last_end:
        cnt += 1
        # Смещаем последнее в предпоследнее и обновляем последнее
        prev_end = last_end
        last_end = end
        # Обнуляем список с началами
        last_starts = [start]
    else:
        last_starts.append(start)

# вывод количества и минимального времени начала рейса,
# который может быть последним
print(cnt, min([i for i in last_starts if i >= prev_end]))
```

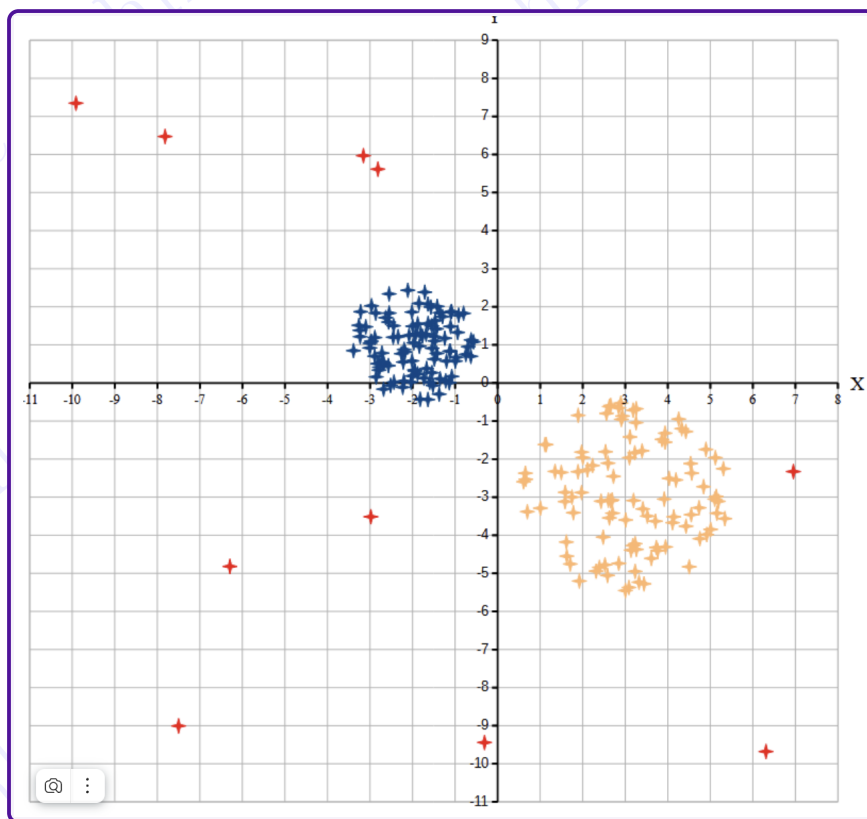
34 Задача 27.1

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике. Каждая звезда обязательно принадлежит только одному из кластеров. Центр кластера, или центроид, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна. Расстояние между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ вычисляется по формуле:

$$d(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Даны два входных файла (файл А и файл Б). В файле А хранятся данные о звёздах двух кластеров. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y (в условных единицах). Известно, что количество звёзд не превышает 1000. В файле Б хранятся данные о звёздах трёх кластеров. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звездах в файле Б аналогична файлу А. Возможные данные одного из файлов иллюстрированы графиком.

Для каждого файла определите координаты центра каждого кластера, затем вычислите четыре числа: для файла А Px – среднее арифметическое абсцисс центров кластеров, и Py – среднее арифметическое ординат центров кластеров; для файла Б Qx – суммы абсцисс центроидов кластеров с минимальным и максимальным размерами (исключая кластер со средним размером), Qy – суммы ординат центроидов кластеров с минимальным и максимальным размерами (исключая кластер со средним размером). В ответе запишите четыре числа: в первой строке сначала целую часть произведения $Px \times 10\,000$, затем целую часть произведения $Py \times 10\,000$ для файла А, во второй строке – сначала целую часть произведения $Qx \times 10\,000$, затем целую часть произведения $Qy \times 10\,000$ для файла Б.



Внимание! График приведён в иллюстративных целях для произвольных значений, не имеющих отношения к заданию. Для выполнения задания используйте данные из прилагаемого файла.

Решение

Для файла A.

```
file = open("...")
clusters = [[] for _ in range(2)]
for star in file:
    x, y = list(map(float, star.strip().split()))
    if -6 * x + 120 < y:
        clusters[0].append((x, y))
    elif y < 15:
        clusters[1].append((x, y))

sum_x = sum_y = 0
for cluster in clusters:
    tx = ty = 0
    mn = 10 ** 20
    for centroid in cluster:
        x1, y1 = centroid
        sm = 0
        for star in cluster:
            x2, y2 = star
            sm += ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
    sum_x += tx
    sum_y += ty
P_x = sum_x / len(clusters)
P_y = sum_y / len(clusters)
print(int(P_x * 10_000), int(P_y * 10_000))
```

Для файла Б.

```
file = open("...")
clusters = [[] for _ in range(3)]
for star in file:
    x, y = list(map(float, star.strip().split()))
    if -6 * x + 120 < y:
        clusters[0].append((x, y))
    elif y < 15:
        clusters[1].append((x, y))
    else:
        clusters[2].append((x, y))

centroids = []
for cluster in clusters:
    tx = ty = 0
    mn = 10 ** 20
    for centroid in cluster:
        x1, y1 = centroid
        sm = 0
        for star in cluster:
            x2, y2 = star
            sm += ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
    centroids.append([len(cluster), tx, ty])

centroids.sort()
Q_x = centroids[0][1] + centroids[2][1]
Q_y = centroids[0][2] + centroids[2][2]
print(int(Q_x * 10_000), int(Q_y * 10_000))
```

35 Задача 27.2

Даны два входных файла (файл А и файл Б). В файле А хранятся данные о звёздах двух кластеров. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y (в условных единицах). Известно, что количество звёзд не превышает 1000. В файле Б хранятся данные о звёздах трёх кластеров. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звездах в файле Б аналогична файлу А. Возможные данные одного из файлов иллюстрированы графиком.

Для файла А определите координаты центра каждого кластера, затем вычислить два числа: Px – минимальное из абсцисс центров кластеров, и Py – минимальное из ординат центров кластеров.

Для файла Б определите координаты центра каждого кластера, затем вычислите два числа: Q_1 – расстояние между центрами кластеров с минимальным и максимальным количеством точек, и Q_2 – максимальное расстояние от центра кластера с минимальным количеством точек до любой точки кластера с максимальным количеством точек.

Гарантируется, что во всех кластерах количество точек различно.

Решение

Для файла А.

```
from math import dist

file = open("...")
clusters = [[] for _ in range(2)]
for star in file:
    x, y = list(map(float, star.strip().split()))
    if -6 * x + 120 < y:
        clusters[0].append((x, y))
    else:
        clusters[1].append((x, y))

centroids = []
min_x, min_y = 0, 0
for cluster in clusters:
    tx = ty = 0
    mn = 10 ** 20
    for centroid in cluster:
        x1, y1 = centroid
        sm = 0
        for star in cluster:
            x2, y2 = star
            sm += dist(centroid, star)
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
    if tx < min_x:
        min_x = tx
    if ty < min_y:
        min_y = ty
print(min_x, min_y)
```

Для файла Б.

```
from math import dist

file = open("...")
clusters = [[] for _ in range(3)]
for star in file:
    x, y = list(map(float, star.strip().split()))
    if -6 * x + 120 < y:
        clusters[0].append((x, y))
    elif y < 15:
        clusters[1].append((x, y))
    else:
        clusters[2].append((x, y))
centroids = []
for cluster in clusters:
    tx = ty = 0
    mn = 10**20
    for centroid in cluster:
        x1, y1 = centroid
        sm = 0
        for star in cluster:
            x2, y2 = star
            sm += dist(centroid, star)
        if sm < mn:
            mn = sm
            tx, ty = x1, y1
    centroids.append([len(cluster), tx, ty, cluster])
centroids.sort()
Q_1 = dist(centroids[0][1:3], centroids[2][1:3])
max_cluster = centroids[2][3]
Q_2 = max([dist(centroids[0][1:3], i) for i in max_cluster])
print(Q_1, Q_2)
```
