

Интеллектуальные робототехнические системы

2022/23 учебный год

Заключительный этап

Предметный тур

Информатика. 8–11 класс

Задача VI.1.1.1. Часы (15 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Часы имеют две кнопки, которые можно использовать для ручной настройки:

- Кнопка А переводит часы вперед на 1 минуту. Например, 05:33 становится 05:34, 16:59 становится 17:00, а 23:59 становится 00:00.
- Кнопка В переводит часы вперед на 1 час. Например, 01:42 становится 02:42, а 23:14 становится 00:14.

Николай заметил, что время на его часах идет неправильно. Он хочет настроить часы на правильное время, нажимая на кнопки как можно меньше раз.

Найдите наименьшее количество нажатий кнопок, необходимое для настройки часов.

Формат входных данных

Первая строка содержит время, отображаемое на часах, в формате `hh:mm` ($00 \leq hh \leq 23$; $00 \leq mm \leq 59$).

Вторая строка содержит правильное время в том же формате.

Формат выходных данных

Выведите одно целое число — наименьшее количество нажатий кнопок, необходимое Николаю, чтобы настроить время на своих часах.

Примеры

Пример №1

Стандартный ввод
11:57 12:00
Стандартный вывод
3

Пример №2

Стандартный ввод
09:09 21:21
Стандартный вывод
24

Пример №3

Стандартный ввод
19:44 08:50
Стандартный вывод
19

Примечание: в первом тестовом примере Николай может настроить время, нажав кнопку А три раза.

Во втором тестовом примере Николай должен нажать на кнопку А 12 раз и кнопку В 12 раз.

Решение

Представим себе окружность, на которой отмечено $24 \cdot 60 = 1440$ точек, и каждая точка соответствует некоторой минуте в сутках. Пронумеруем точки по часовой стрелке от 0 до 1439, где точка 0 — это время 00:00, а точка 1439 — это время 23:59. Пусть время на часах соответствует точке x , а правильное время — точке y .

Заметим, что кнопка А увеличивает номер текущей точки на 1, а кнопка В — на 60.

Найдём d — расстояние между точками по часовой стрелке. Если $x \leq y$, то $d = y - x$, иначе $d = (y + 1440) - x$.

Мы хотим как можно быстрее продвинуться на расстояние d , используя прыжки вперёд на расстояния 1 и 60. Легко понять, что нам нужно использовать $d \bmod 60$ прыжков первого типа и $d \operatorname{div} 60$ прыжков второго типа. Значит, ответ на задачу равен $d \bmod 60 + d \operatorname{div} 60$.

Можно было и не использовать при решении задачи конструкцию с окружностью, описанную выше, а просто промоделировать процесс, если догадаться, что сначала нужно увеличивать минуты, пока они не станут правильными, а потом — увеличивать часы.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 def t():
2     s = input()
3     return int(s[0:2]) * 60 + int(s[3:5])
4
5 t1, t2 = t(), t()
6 d = t2 - t1
7 if d < 0:
8     d += 1440
9 print(d % 60 + d // 60)
```

Задача VI.1.1.2. Калькулятор (18 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Вам дано число. Выведите, как это число будет выглядеть на экране калькулятора. Гарантируется, что вводимое число неотрицательное, строго меньше 10^{500} . Цифры должны быть символьными прямоугольниками 7×5 , такими как показано ниже.

```
....*   *****   *****   *...*   *****   *****   *****   *****   *****   *****
....*   ....*   ....*   *...*   *....   *....   ....*   *...*   *...*   *...*
....*   ....*   ....*   *...*   *....   *....   ....*   *...*   *...*   *...*
....*   *****   *****   *****   *****   *****   ....*   *****   *****   *...*
....*   *....   ....*   ....*   ....*   *...*   ....*   *...*   ....*   *...*
....*   *....   ....*   ....*   ....*   *...*   ....*   *...*   ....*   *...*
....*   *****   *****   ....*   *****   *****   ....*   *****   *****   *****
```

Соседние цифры следует разделять столбцами 7×1 , заполненными точками.

Формат входных данных

В единственной строке входных данных находится целое число n , которое надо вывести на экран калькулятора ($0 \leq n < 10^{500}$).

Формат выходных данных

Выведите число n так, как показано в примере.

Примеры

Пример №1

Стандартный ввод
123
Стандартный вывод
....*.*****.******.....*.....**.....*.....**.*****.******. *.....**. *.....**.*****.*****

Решение

В этой задаче требовалось сделать ровно то, что написано в условии. Сначала вывести первую строчку всех цифр, потом вторую и так далее. Чтобы это было проще сделать, можно заметить, что есть всего 4 типа различных строчек, которые бывают во всех цифрах. А именно строчки «*....», «....*», «*...*» и «*****». Поэтому для каждой из цифр можно было сделать массив из 7-ми чисел, определяющие, какая из этих строчек первая строчка цифры, вторая и так далее. И тогда для каждой из 7-ми строчек вывода можно просто выводить одну из возможных строчек для каждой цифры числа.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1  arr = ['. ....', '....*', '*.....', '*...*', '*****'];
2  mas = [[4, 3, 3, 3, 3, 3, 4],
3         [1, 1, 1, 1, 1, 1, 1],
4         [4, 1, 1, 4, 2, 2, 4],
5         [4, 1, 1, 4, 1, 1, 4],
6         [3, 3, 3, 4, 1, 1, 1],
7         [4, 2, 2, 4, 1, 1, 4],
8         [4, 2, 2, 4, 3, 3, 4],
9         [4, 1, 1, 1, 1, 1, 1],
10        [4, 3, 3, 4, 3, 3, 4],
11        [4, 3, 3, 4, 1, 1, 4]]
12
13  v = input()
14  n = len(v)
15  for i in range(7):
16      for j in range(n):
17          if (j > 0):
18              print('.', end = '')
19              print(arr[mas[int(v[j])][i]], end = '')
20  print('')
```

Задача VI.1.1.3. «Формула 2» (20 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Гонки «Формулы 2» — популярный вид развлечений. В ней участвуют не машины, а андроиды. У всех участников на спине нарисован номер от 1 до n , номер у каждого участника уникален.

Очень хитрый андроид АНК решил продвинуть свою команду на первые места, поэтому всех участников, у которых номер не превосходит число k (это вся его команда) он напоил ультрасовременным топливом. Любой андроид, в котором есть такое топливо гарантированно быстрее андроида, в котором такого топлива нет. Теперь вся его команда каким-то образом поделит первые k позиций итоговой таблицы результатов.

Ваша задача — определить, сколько андроидов могло быть в команде АНК. Нужно найти все возможные значения k . Вам дана только таблица результатов гонки.

Формат входных данных

В первой строке содержится одно целое число n ($1 \leq n \leq 200\,000$) — количество андроидов, которые приняли участие в гонках «Формулы 2». Вторая строка содержит n различных чисел a_i ($1 \leq a_i \leq n$) — таблица результатов гонки, где a_i — номер на спине андроида, который занял i -е место.

Формат выходных данных

В первой строке выведите одно целое число s — количество возможных значений числа k . В следующей строке выведите s натуральных чисел, разделенных пробелами — возможные значения числа k . Все числа во второй строке должны быть различны и не должны превосходить n . Числа во второй строке должны быть упорядочены по возрастанию.

Примеры

Пример №1

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

Решение

Переформулируем задачу: требуется найти все такие k , что среди первых k элементов нет таких, которые больше чем k , то есть встречаются все числа от 1 до k . Чтобы решить эту задачу, будем просматривать массив и поддерживать максимальное число на префиксе. Если максимальное число на префиксе равно длине префикса, то это означает что все числа на префиксе не больше k , следовательно, это ответ.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n = int(input())
2 a = list(map(int, input().split()))
3 ans = []
4 ma = 0
5 for i in range(n):
6     ma = max(ma, a[i])
7     if i + 1 == ma:
8         ans.append(i + 1)
9 print(len(ans))
10 print(*ans)
```

Задача VI.1.1.4. Подпалиндромы (22 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Найти количество подпоследовательностей длины три, состоящих из букв, которые являются палиндромами.

Последовательность $Z = z_1, z_2, \dots, z_k$ является подпоследовательностью последовательности $X = x_1, x_2, \dots, x_m$, если существует строго возрастающая последовательность i_1, i_2, \dots, i_k индексов X таких, что для всех $j = 1, 2, \dots, k$ выполняется соотношение $x_{i_j} = z_j$.

Другими словами, подпоследовательность данной последовательности — это последовательность, из которой удалили ноль или больше элементов. Например, $Z = bcd$ является подпоследовательностью последовательности $X = abcdbdab$, а соответствующая последовательность индексов имеет вид 2, 3, 5, 7.

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, $abba$ — палиндром, а $otax$ — нет.

Формат входных данных

Вам дана строка, длина которой не превышает $3 \cdot 10^5$. Гарантируется, что строка может содержать только строчные буквы английского алфавита и пробелы. Также

гарантируется, что строка не пустая. Строка не может начинаться с пробела или заканчиваться им.

Формат выходных данных

Выведите одно число — количество палиндромов длины три, встречающихся как подпоследовательность.

Примеры

Пример №1

Стандартный ввод
abracadabra
Стандартный вывод
50

Пример №2

Стандартный ввод
abra ca dabra
Стандартный вывод
50

Решение

Удалим все пробелы, так как они не влияют на количество палиндромов.

Переберем центр палиндрома. Тогда первая буква будет совпадать с последней. Переберем, какой она может быть от *a* до *z*. Чтобы узнать сколько у нас палиндромов с таким центром и зафиксированной буквой, нужно знать, сколько раз встречается буква левее центра палиндрома и сколько раз правее и перемножить эти количества. Для того чтобы узнать сколько раз встречается левее и сколько раз правее, воспользуемся техникой частичных сумм.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 import string
2
3 s = input().replace(" ", "")
4 n = len(s)
5 cnt = [[0]* 26 for _ in range(n + 1)]
6
7 for i in range(n):
8     cnt[i + 1] = cnt[i].copy()
9     cnt[i + 1][ord(s[i]) - 97] += 1
10 ans = 0
11 for i in range(1, n - 1):
```

```

12     for c in range(0, 26):
13         ans += cnt[i][c] * (cnt[n][c] - cnt[i + 1][c]);
14
15     print(ans)

```

Задача VI.1.1.5. Самый длинный путь (25 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Вам дана матрица n на m натуральных чисел. Ваша задача — найти в ней путь максимальной длины. Путь — последовательность подряд идущих натуральных чисел, стоящих в соседних по стороне клетках.

Формат входных данных

В первой строке заданы числа n и m ($1 \leq n, m \leq 1000$). Далее в n строках задано по m чисел $a_{i,j}$ — элементы матрицы ($1 \leq a_{i,j} \leq 10 \times n \times m$).

Формат выходных данных

Выведете максимальную длину пути в матрице.

Примеры

Пример №1

Стандартный ввод
3 3 1 2 3 6 5 4 7 8 9
Стандартный вывод
9

Пример №2

Стандартный ввод
3 3 1 1 1 1 2 1 1 1 1
Стандартный вывод
2

Решение

Решим задачу динамическим программированием. $dp[i][j]$ — максимальная длина пути, оканчивающаяся в клетке (i, j) .

Чтобы посчитать $dp[i][j]$, нам нужно знать $dp[i-1][j]$, $dp[i+1][j]$, $dp[i][j-1]$, $dp[i][j+1]$, если значения соседа в исходной матрицы меньше нас на один, то обновляем текущее значение динамики. Для пересчёта значения динамики воспользуемся техникой ленивой динамики, чтобы не задумываться о порядке обхода. Ответом будет самое большое значение $dp[i][j]$.

Пример программы-решения

Ниже представлено решение на языке C++.

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4
5  using namespace std;
6
7  int n, m;
8  vector<vector<int>> a, dp;
9
10 int get(int i, int j)
11 {
12     if (dp[i][j] != -1)
13         return dp[i][j];
14     int best = 1;
15     if (i != 0 && a[i - 1][j] == a[i][j] + 1)
16         best = max(best, get(i - 1, j) + 1);
17     if (i != n - 1 && a[i + 1][j] == a[i][j] + 1)
18         best = max(best, get(i + 1, j) + 1);
19     if (j != 0 && a[i][j - 1] == a[i][j] + 1)
20         best = max(best, get(i, j - 1) + 1);
21     if (j != m - 1 && a[i][j + 1] == a[i][j] + 1)
22         best = max(best, get(i, j + 1) + 1);
23     return dp[i][j] = best;
24 }
25
26 int main()
27 {
28     cin >> n >> m;
29     a = vector<vector<int>>(n, vector<int>(m));
30     dp = vector<vector<int>>(n, vector<int>(m, -1));
31     for (int i = 0; i < n; i++)
32         for (int j = 0; j < m; j++)
33             cin >> a[i][j];
34     int ans = 1;
35     for (int i = 0; i < n; i++)
36         for (int j = 0; j < m; j++)
37             ans = max(ans, get(i, j));
38     cout << ans << "\n";
39 }
```

Математика. 8–9 классы

Задача VI.1.2.1. (15 баллов)

Темы: комбинаторика, основная теорема арифметики.

Условие

Сколько различных делителей у числа 1800?

Решение

Разложим число 1800 на множители. Имеем:

$$1800 = 2^3 \cdot 3^2 \cdot 5^2.$$

Введем в рассмотрение тройки (a, b, c) , где $a \in \{0, 1, 2, 3\}$, $b, c \in \{0, 1, 2\}$. Делителем, отвечающим тройке (a, b, c) , назовем число

$$2^a \cdot 3^b \cdot 5^c.$$

Ясно, что каждому делителю числа 1800 соответствует ровно одна тройка (a, b, c) , и наоборот. Тем самым, достаточно посчитать количество различных (упорядоченных) троек. Их будет $4 \cdot 3 \cdot 3 = 36$.

Ответ: 36.

Критерии оценивания

- Выполнено разложение числа на множители — 2 балла.
- Установлена связь между наборами степеней и количеством делителей — 6 баллов.
- Предпринята попытка нахождения количества троек, но неверно установлен их тип — 8 баллов.
- При подсчете количества троек допущена арифметическая ошибка — 13 баллов.

Задача VI.1.2.2. (20 баллов)

Темы: теория чисел, десятичная система счисления.

Условие

Взяли три различные цифры a, b, c , отличные от 0. Оказалось, что сумма всевозможных трёхзначных чисел, образованных из этих цифр, равна утроенному трёхзначному числу, составленному только из цифры a . Найдите все такие цифры a, b, c .

Решение

Обозначая через \overline{xyz} трёхзначное число с цифрами x, y, z , запишем равенство

$$\overline{abc} + \overline{acb} + \overline{bac} + \overline{bca} + \overline{cab} + \overline{cba} = 3\overline{aaa}.$$

Учитывая, что $\overline{xyz} = 100x + 10y + z$, записанное выше равенство равносильно

$$222a + 222b + 222c = 333a \iff 2(b + c) = a.$$

Значит, число a — четное.

Поскольку b и c не могут одновременно быть равными 1, то $b + c > 2$. Следовательно, $a > 4$.

Рассмотрим два случая:

1. $a = 6$. Тогда $b + c = 3$, откуда $b = 1, c = 2$ или $b = 2, c = 1$.
2. $a = 8$. Тогда $b + c = 4$, откуда $b = 1, c = 3$ или $b = 3, c = 1$.

Ответ: 4 тройки (a, b, c) : $(6, 1, 2)$, $(6, 2, 1)$, $(8, 1, 3)$, $(8, 3, 1)$.

Критерии оценивания

- Записаны все варианты трехзначных чисел — 2 балла.
- Числа представлены в виде суммы разрядных единиц — 2 балла (прибавляется).
- Выписано и преобразовано уравнение для цифр — 8 баллов.
- Найдены оценки для цифр — 12 баллов.
- Получен какой-то один вариант решения — 15 баллов.

Задача VI.1.2.3. (20 баллов)

Темы: алгебра, система уравнений, текстовая задача.

Условие

Пользователь смартфона знает, что его устройство в обычном режиме расходует полный заряд батареи ровно за 24 ч, а в режиме энергосбережения за каждые 40 мин расходуется на 1% заряда меньше, чем в обычном режиме. Пользователь заметил, что на его смартфоне осталось 16% заряда, а возможность зарядить устройство появится только через 4,5 ч. Всё это время пользователю нужно быть на связи. Какое максимальное время ещё можно использовать смартфон в обычном режиме?

Решение

Расход заряда смартфона в процентах за минуту равен

$$r = \frac{100}{24 \cdot 60} = \frac{5}{72}.$$

За 40 минут устройство тратит $40r\%$ заряда батареи. Значит, в режиме энергосбережения тратится $40r - 1\%$ за 40 минут. То есть расход в этом режиме составляет:

$$r_e = \frac{40r - 1}{40} = r - \frac{1}{40}$$

процентов заряда в минуту.

Если устройство работает t минут в обычном режиме и t_e — в энергосберегающем, то израсходуется $rt + r_e t_e$ процентов заряда батареи. По условию задачи должно быть

$$\begin{cases} t + t_e = 60 \cdot 4,5 = 270 \text{ мин,} \\ rt + r_e t_e = 16\%. \end{cases}$$

Следовательно,

$$rt + r_e(270 - t) = 16.$$

Отсюда

$$t = \frac{16 - 270r_e}{r - r_e} = 40 \cdot \left(16 - 270 \cdot \left(\frac{5}{72} - \frac{1}{40} \right) \right) = 160 \text{ мин.}$$

Ответ: 160 мин (2 ч 40 мин).

Критерии оценивания

- Установлена зависимость в виде формулы между расходами заряда в обычном и энергосберегающем режиме — 4 балла.
- Составлена система, которой удовлетворяет искомое время — 10 баллов.
- Допущена арифметическая ошибка при условии, что всё остальное верно — 18 баллов.

Задача VI.1.2.4. (20 баллов)

Темы: графы, логика.

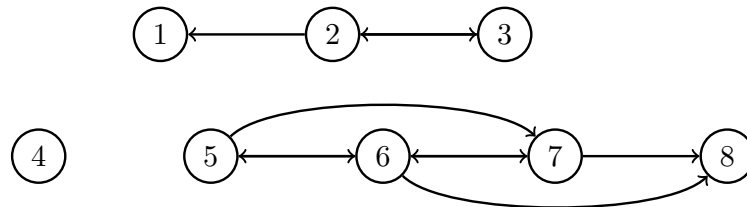
Условие

Населённый пункт состоит из двух районов, расположенных по двум берегам реки и соединённых мостом. На левом берегу есть три бензоколонки (1, 2, 3), а на правом берегу — пять (4, 5, 6, 7, 8). На данный момент мост ремонтируют, и проезд по нему возможен только в одну сторону. Тем не менее, существует и при том единственный путь, по которому бензовоз может объехать все заправок населённого пункта, побывав на каждой лишь один раз. В таблице указана информация о дорогах между бензоколонками (если из i -й есть дорога в j -ю, то на пересечении i -й строки и j -го столбца стоит 1, а если дороги нет, то 0). Опишите все возможные способы заполнить 4-ю строку таблицы.

	1	2	3	4	5	6	7	8
1	0	0	0	*	0	0	0	0
2	1	0	1	*	0	0	0	0
3	0	1	0	*	0	0	0	0
4	*	*	*	0	*	*	*	*
5	0	0	0	*	0	1	1	0
6	0	0	0	*	1	0	1	1
7	0	0	0	*	0	1	0	1
8	0	0	0	*	0	0	0	0

Решение

Изобразим граф, для которого данная в условии таблица является матрицей смежности.



Назовём «путём бензовоза» тот путь, по которому можно объехать все заправки, заехав на каждую лишь один раз. Обозначим через m_j элемент таблицы на пересечении 4-й строки и j -го столбца. По условию движение по мосту возможно лишь в одном направлении. Рассмотрим два случая.

Первый случай: мост ведёт с правого берега на левый. Тогда путь бензовоза начинается на правом берегу и заканчивается на левом. Мост необходимо ведёт к бензоколонке 3. Следовательно, $m_3 = 1$.

Путь бензовоза на правом берегу должен заканчиваться на бензоколонке 4. Колонка 8 — не последняя на пути, но из неё не ведёт путь в 5, 6, 7 и к колонкам другого берега, поэтому должен быть путь из 8 в 4. Теперь мы получили противоречие, существует два пути бензовоза: 5, 6, 7, 8, 4, 3, 2, 1 и 5, 7, 6, 8, 4, 3, 2, 1.

Второй случай: мост ведёт с левого берега на правый. Тогда $m_1 = m_2 = m_3 = 0$.

Путь бензовоза начинается на левом берегу и заканчивается на правом. Мост необходимо соединяет бензоколонки 1 и 4.

Пути от 4-й к 5-й бензоколонке быть не может, иначе получается два пути бензовоза. Поэтому $m_5 = 0$.

Путь бензовоза не может идти от 4-й к 7-й или от 4-й к 8-й бензоколонке, поэтому необходимо $m_6 = 1$. Вне зависимости от значений m_7 и m_8 в этом случае получается единственный путь 3, 2, 1, 4, 6, 5, 7, 8.

Ответ: существует 4 способа заполнить 4-ю строку: 000001**, где * — число 1 или 0.

Критерии оценивания

- Перечислены лишь некоторые способы заполнить 4-ю строку — 5 баллов.

- Определены лишь некоторые символы 4-й строки — 3 балла.

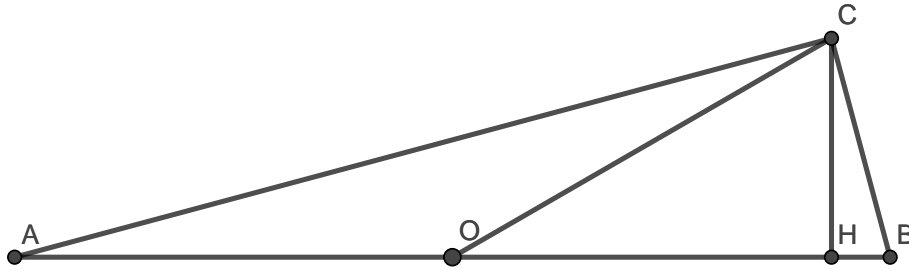
Задача VI.1.2.5. (25 баллов)

Темы: планиметрия, геометрическая вероятность.

Условие

Треугольник ABC , один из углов которого равен 15° , вписан в окружность так, что одна из его сторон лежит на диаметре. Высота треугольника, опущенная на диаметр, делит его на два треугольника. Найти вероятность того, что точка, наудачу брошенная в треугольник ABC , попадет в меньший из треугольников.

Решение



Поскольку одна из сторон треугольника ABC совпадает с диаметром описанной окружности, то данный треугольник — прямоугольный.

Пусть H — основание высоты треугольника ABC , проведённой из вершины прямого угла C , $\angle CAB = 15^\circ$, O — центр описанной окружности, которая имеет радиус r .

Поскольку угол COB опирается на ту же дугу, что и угол CAB , то $\angle COB = 2\angle CAB = 30^\circ$. Поэтому катет CH в прямоугольном треугольнике COH равен половине гипотенузы CO , которая является радиусом описанной окружности. Таким образом, $CH = r/2$.

Заметим, что треугольники ACH и BCH подобны. Отсюда следует, что $CH : AH = BH : CH$, то есть

$$\frac{r/2}{2r - x} = \frac{x}{r/2},$$

где $x = BH$. Получаем уравнение для величины x :

$$x^2 - 2rx + \frac{r^2}{4} = 0 \iff x = r \pm r \frac{\sqrt{3}}{2}.$$

Так как угол COB острый, то точка H лежит правее O , значит, $x < r$. Поэтому

$$x = r \left(1 - \frac{\sqrt{3}}{2} \right).$$

Наудачу брошенная в треугольник ABC точка попадёт в треугольник BCH с вероятностью p , равной отношению площадей треугольников BCH и ABC :

$$p = \frac{1/2 \cdot CH \cdot HB}{1/2 \cdot CH \cdot AB} = \frac{x}{2r} = \frac{1}{2} \left(1 - \frac{\sqrt{3}}{2} \right) = \frac{2 - \sqrt{3}}{4}.$$

Поскольку $p < 1/2$, то треугольник BCH по площади меньше треугольника ACH , поэтому вероятность p — искомая.

Ответ: $\frac{2 - \sqrt{3}}{4}$.

Критерии оценивания

- Указано, что отрезок OC равен половине гипотенузы — 4 балла.
- Найдена величина высоты CH — 4 балла.
- Выписаны соотношения, вытекающие из подобия треугольников, на которые высота CH разбивает треугольник ABC — 2 балла.
- Отмечено и обосновано, что вероятность равна отношению площадей треугольников BCH и ABC — 2 балла.

Математика. 10–11 классы

Задача VI.1.3.1. (15 баллов)

Темы: диофантово уравнение, делимость чисел.

Условие

Решите уравнение $x^2 + 5xy - 6y^2 + x + 6y = 2023$ в целых неотрицательных числах.

Решение

Рассмотрим левую часть уравнения как квадратный трёхчлен относительно переменной x :

$$x^2 + 5xy - 6y^2 + x + 6y = x^2 + x(5y + 1) - 6y^2 + 6y.$$

Его дискриминант

$$D = (5y + 1)^2 - 4(-6y^2 + 6y) = 49y^2 - 14y + 1 = (7y - 1)^2.$$

Корни

$$x = \frac{-(5y + 1) \pm (7y - 1)}{2} = y - 1; -6y$$

дают разложение на множители

$$x^2 + 5xy - 6y^2 + x + 6y = (x - (y - 1))(x - (-6y)) = (x - y + 1)(x + 6y).$$

Так как $2023 = 7 \cdot 17^2$, то исходное уравнение равносильно уравнению

$$(x - y + 1)(x + 6y) = 7 \cdot 17^2.$$

По условию числа x и y целые, $x, y \geq 0$. Получаем следующие варианты:

$$\begin{cases} x - y + 1 = 1 \\ x + 6y = 7 \cdot 17^2 \end{cases} \quad (1) \quad \begin{cases} x - y + 1 = 7 \\ x + 6y = 17^2 \end{cases} \quad (2) \quad \begin{cases} x - y + 1 = 17 \\ x + 6y = 7 \cdot 17 \end{cases} \quad (3)$$
$$\begin{cases} x - y + 1 = 7 \cdot 17 \\ x + 6y = 17 \end{cases} \quad (4) \quad \begin{cases} x - y + 1 = 17^2 \\ x + 6y = 7 \end{cases} \quad (5) \quad \begin{cases} x - y + 1 = 7 \cdot 17^2 \\ x + 6y = 1 \end{cases} \quad (6)$$

Заметим, что $x - y \leq x + 6y$. Это неравенство позволяет сразу исключить случаи (4), (5), (6).

В каждой из систем (1), (2), (3) вычтем из второго уравнения первое. Имеем

(1): $7y = 7 \cdot 17^2$, значит $y = 17^2 = 289$, $x = 289$.

(2): $7y = 17^2 - 6 = 283$. Число 283 не делится на 7, поэтому целых решений нет.

(3): $7y = 7 \cdot 17 - 16$. Правая часть не делится на 7, поэтому целых решений нет.

Ответ: $x = 289, y = 289$.

Критерии оценивания

- Выполнено разложение многочлена и числа в правой части уравнения на множители — 5 баллов.
- Рассмотрены некоторые варианты систем — 9 баллов.

Задача VI.1.3.2. (20 баллов)

Темы: графы, комбинаторика.

Условие

В вершинах графа, содержащего 2023 вершины, расставлены некоторые целые числа. Между двумя вершинами проведено ребро, если и только если числа в этих вершинах в сумме дают чётное число. Какое наименьшее количество рёбер может быть в таком графе?

Решение

Заметим, что в данном графе каждое чётное число будет соединено со всеми остальными чётными, а каждое нечётное — со всеми остальными нечётными. Таким образом, граф составлен из двух полных подграфов.

Полный граф из n вершин содержит $C_n^2 = \frac{n(n-1)}{2}$ рёбер (число неупорядоченных пар, которые можно образовать из n элементов). Если среди 2023 чисел будет 1011 чётных и 1012 нечётных, то количество рёбер в таком графе

$$C_{1011}^2 + C_{1012}^2 = \frac{1011 \cdot 1010}{2} + \frac{1012 \cdot 1011}{2} = 1011^2.$$

Допустим, в графе $1011 - k$ чётных и $1012 + k$ нечётных чисел, где $k \in \mathbb{N}$. Тогда количество рёбер в таком графе

$$\begin{aligned} C_{1011-k}^2 + C_{1012+k}^2 &= \frac{(1011 - k)(1010 - k)}{2} + \frac{(1012 + k)(1011 + k)}{2} \\ &= 1011^2 + k^2 + k > 1011^2. \end{aligned}$$

Следовательно, число рёбер будет наименьшим и равным 1011^2 , если в вершинах графа записано 1011 чётных и 1012 нечётных чисел (или наоборот).

Ответ: $1011^2 = 1022121$.

Критерии оценивания

- Отмечено, что граф составлен из двух полных подграфов — 4 балла.
- Вычислено количество рёбер в графе, содержащем 1011 чётных (нечётных) и 1012 нечётных (чётных) чисел — 4 балла.

Задача VI.1.3.3. (20 баллов)

Темы: текстовая задача, уравнение с параметром.

Условие

Игрок №7 футбольной команды, находясь на оси Ox слева от точки $x = 0$, хочет передать пас. Если мяч попадёт в точку отрезка $[2, 4]$ оси Ox , то пас сможет принять игрок №9. В точке $x = 0$ оказывается игрок команды соперников, имеющий рост 2 вдоль оси Oy и умеющий подпрыгивать на любую высоту, меньшую, чем 0,5. Игрок №7 посылает мяч по траектории $y = -\frac{x^2}{4} - \frac{x}{2} + c$, выбирая параметр c произвольно. Определите все отрицательные точки оси Ox , находясь в которых игрок №7 сможет передать мяч игроку №9.

Решение

Обозначим через x_1 координату по оси Ox игрока №7, а через x_2 — координату по оси Ox точки, в которую прилетит переданный мяч при ударе о землю. Тогда x_1, x_2 — это корни уравнения

$$-\frac{x^2}{4} - \frac{x}{2} + c = 0.$$

По теореме Виета имеем

$$\begin{cases} -(x_1 + x_2) = \frac{-1/2}{-1/4} = 2, \\ x_1 x_2 = \frac{c}{-1/4} = -4c. \end{cases}$$

Чтобы игрок №9 смог принять пас, необходимо, во-первых, чтобы $x_2 \in [2, 4]$, во-вторых, игрок команды соперников не должен перехватить мяч. Для этого необходимо $[-x^2/4 - x/2 + c]_{x=0} \geq 2 + 0,5 = 2,5$. То есть должно быть $c \geq 2,5$.

Тогда из первого уравнения системы получаем $x_1 = -2 - x_2 \in [-6, -4]$. Из второго уравнения находим

$$x_1 x_2 = x_1(-2 - x_1) = -4c \leq -4 \cdot 2,5 = -10.$$

Следовательно, число x_1 удовлетворяет неравенству

$$x_1^2 + 2x_1 - 10 \geq 0.$$

Отсюда, учитывая, что $x_1 \leq 0$, находим $x_1 \in (-\infty, -\sqrt{11} - 1]$.

$$\text{Тогда } x_1 \in [-6, -4] \cap (-\infty, -\sqrt{11} - 1] = [-6, -\sqrt{11} - 1].$$

Ответ: $[-6, -\sqrt{11} - 1]$.

Критерии оценивания

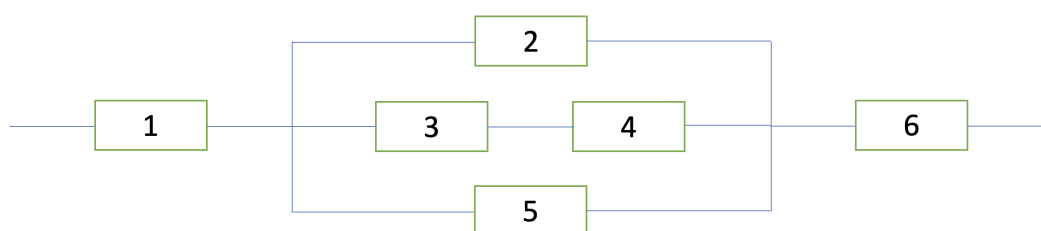
- Получено условие $x_1 \in [-6, -4]$ — 3 балла.
- Получена оценка $x_1 \leq -\sqrt{11} - 1$ — 5 баллов.
- Допущена несущественная арифметическая ошибка, при условии, что всё остальное верно — 17 баллов.

Задача VI.1.3.4. (20 баллов)

Темы: теория вероятностей, теоремы сложения и умножения.

Условие

Перед вами система труб, в которой находятся предохранительные ворота A_1, A_2, \dots, A_6 .



Вероятность того, что ворота A_i закроются при попадании огня в систему, равна $p_i = \frac{i}{i+2}$. Маг использует заклинание «Огненное дыхание» на системе труб. Какова вероятность, что огонь пройдет через систему, если считается, что ворота закрываются независимо?

Решение

Для того чтобы огонь прошел через систему, он должен пройти через ворота A_1 и A_6 , а также через «средний блок». Первые две вероятности равны $(1 - p_1)$ и $(1 - p_6)$, соответственно.

Вычислим вероятность пройти через «средний блок». Для этого найдем вероятность не пройти через него и вычтем ее из 1. Не пройти через блок означает не пройти ни по одной из ниток блока. Вероятность не пройти по верхней нитке, через блок A_2 , равна p_2 . Вероятность не пройти по нижней нитке, через блок A_5 , равна p_5 . Вероятность не пройти через нить, проходящую через блоки A_3 и A_4 , может быть вычислена с использованием противоположного события – «пройти по нити», и равна $1 - (1 - p_3)(1 - p_4)$. Итого,

$$\begin{aligned} & (1 - p_1)(1 - p_6)(1 - p_2p_5(1 - (1 - p_3)(1 - p_4))) = \\ & = \left(1 - \frac{1}{3}\right) \left(1 - \frac{3}{4}\right) \left(1 - \frac{1}{2} \cdot \frac{5}{7} \left(1 - \left(1 - \frac{3}{5}\right) \left(1 - \frac{2}{3}\right)\right)\right) = \frac{29}{252}. \end{aligned}$$

Ответ: $\frac{29}{252}$.

Критерии оценивания

- Предложено правильное использование теорем сложения и умножения на всей системе труб — 8 баллов.
- Правильно используются вероятности противоположных событий — 6 баллов (суммируются с предыдущим).
- Получен неверный ответ из-за арифметической ошибки — 18 баллов.

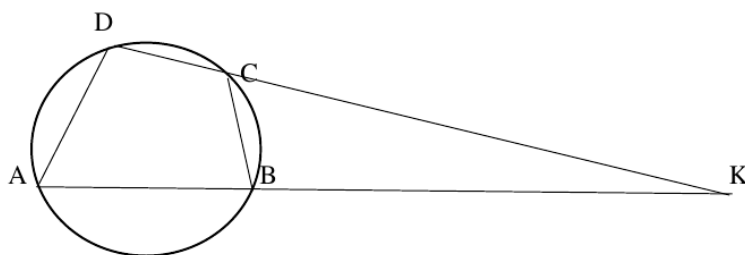
Задача VI.1.3.5. (25 баллов)

Темы: планиметрия, геометрическая вероятность.

Условие

В окружность вписан четырехугольник $ABCD$. Прямые, на которых лежат стороны AB и DC , пересекаются в точке K . Известно, что $\cos \angle AKD = \frac{2\sqrt{2}}{3}$, $CK = 9$, а отрезок AK в 3 раза больше стороны AD . Вероятность того, что точка, случайным образом брошенная в треугольник AKD , попадет в четырехугольник $ABCD$, равна $\frac{5}{9}$. Найдите длину стороны DC .

Решение



1) В любом вписанном четырехугольнике сумма противоположных углов равна:

$$180^\circ \implies \angle ADC + \angle ABC = 180^\circ \implies \angle ABC = 180^\circ - \angle ADC$$

С другой стороны,

$$\angle CBK = 180^\circ - \angle ABC \implies \angle CBK = 180^\circ - (180^\circ - \angle ADC) = \angle ADC.$$

2) Рассмотрим треугольники CBK и AKD : угол AKD у них общий, $\angle CBK = \angle ADC$, поэтому они подобны, следовательно,

$$\frac{BC}{AD} = \frac{CK}{AK} \implies \frac{BC}{AD} = \frac{9}{3AD} \implies BC = 3.$$

3) Рассмотрим треугольник CBK : $BC = 3$, $CK = 9$, $\cos \angle AKD = \frac{2\sqrt{2}}{3}$. Для того чтобы найти сторону BK , воспользуемся теоремой косинусов:

$$\begin{aligned} BC^2 &= BK^2 + CK^2 - 2BK \cdot CK \cdot \cos \angle AKD \implies 3^2 = BK^2 + 9^2 - 2BK \cdot 9 \cdot \frac{2\sqrt{2}}{3} \\ &\implies BK^2 - 12\sqrt{2}BK + 72 = 0 \implies BK = 6\sqrt{2}. \end{aligned}$$

4) По условию вероятность того, что точка, случайным образом брошенная в треугольник AKD , попадет в четырехугольник $ABCD$, равна $\frac{5}{9}$. Событие, состоящее в том, что точка, случайным образом брошенная в треугольник AKD , попадет в треугольник CBK , является противоположным по отношению к событию, что она попадет в четырехугольник $ABCD$. Следовательно, вероятность того, что точка, случайным образом брошенная в треугольник AKD , попадет в треугольник CBK , будет равна $1 - \frac{5}{9} = \frac{4}{9}$.

С другой стороны, вероятность того, что точка, случайным образом брошенная в треугольник AKD , попадет в треугольник CBK , будет равна отношению их площадей, следовательно,

$$\frac{S_{\Delta CBK}}{S_{\Delta AKD}} = \frac{4}{9}.$$

5) Так как треугольники CBK и AKD подобны, отношение их площадей равно квадрату коэффициента подобия k , поэтому

$$k^2 = \frac{S_{\Delta CBK}}{S_{\Delta AKD}} = \frac{4}{9} \implies k = \frac{2}{3} \implies \frac{BK}{DK} = \frac{2}{3} \implies DK = \frac{3}{2}BK = \frac{3}{2} \cdot 6\sqrt{2} = 9\sqrt{2}.$$

$$\text{Значит, } DC = DK - CK = 9\sqrt{2} - 9 = 9(\sqrt{2} - 1).$$

Ответ: $9(\sqrt{2} - 1)$.

Критерии оценивания

- Указано, что в любом вписанном четырехугольнике сумма противоположных углов равна 180° — 4 балла.
- Доказано, что треугольники CBK и AKD подобны — 5 баллов.
- Найден отрезок BK — 5 баллов.
- Найден коэффициент подобия треугольников CBK и AKD — 4 балла.

Приложения ко второму этапу

Технические особенности и компиляция кода

- Решения тестируются на виртуальной машине в Azure под управлением ОС CoreOS (x64): 4vCPU Intel(R) Xeon(R) E5-2673 v3 2.40GHz, 8GB RAM;
- Существует ограничение на размер кода решения — 10 МВ;
- Существуют ограничения на максимальный объем памяти и процессорное время для компиляции решения равные 512 МВ и 30 секунд соответственно.

Для компиляции и выполнения решений используется следующее ПО

- C++ (g++ 6.3.0)
 - Compile command: `g++ -pipe -O2 -static -o main`
- C++11 (g++ 6.3.0)
 - Compile command: `g++ -std=c++11 -pipe -O2 -static -o main`
- C++20 (g++ 10.2.1)
 - Compile command: `g++ -std=c++20 -pipe -O2 -static -o main`

*Python 3.6.**

```
Cython==0.29.21
django==3.2.3
hypothesis==3.4.2
lxml==3.6.0
matplotlib==2.2.3
mysqlclient==1.3.13
networkx==2.4
nltk==3.2.2
numpy==1.14.1
opencv-python-headless==3.4.4
pandas==1.0.5
pymorphy2==0.8
pymystem3==0.1.5
pytest==2.5.2
requests==2.24.0
rsa==3.4.2
scikit-image==0.13.1
scikit-learn==0.19.1
scipy==1.0.0
sumy==0.6.0
sympy==1.4
toolz==0.8.2
torch==0.3.1
```

torchvision==0.2.0

*Python 3.10.**

apache-airflow==2.2.3
beautifulsoup4==4.10.0
Cython==0.29.26
django==4.0.1
django-rest-framework==3.13.1
factory_boy==3.2.1
fasttext==0.9.2
gensim==4.1.2
hypothesis==6.36.0
lxml==4.7.1
matplotlib==3.5.1
mip==1.13.0
myhdl==0.11
mysqlclient==2.1.0
networkx==2.6.3
nltk==3.6.7
numpy==1.22.1
opencv-python-headless==4.5.5.62
pandas==1.4.0
pycosat==0.6.3
pymorphy2==0.9.1
pymystem3==0.2.0
pysqlite3-binary==2.6.0
pytest==6.2.5
requests==2.27.1
rsa==4.8
scikit-image==0.19.1
scikit-learn==1.0.2
scipy==1.7.3
shapely==1.8.0
sumy==0.9.0
sympy==1.9
toolz==0.11.2

Инструкция по отправке маркеров и коммуникации с системой проверки

На гите доступны методы `getTask()` и `sendMessage()` для класса `Task`. Но, они добавлены только для того, чтобы ваше решение могло компилироваться у вас на компьютере. У этих методов не написана реализация, на вашем компьютере они ничего делать не будут, но будут работать на тестирующей системе.

Для задачи IV.1

Нужен только метод `getTask()`, который вернет одно число в виде строки. Вызывать этот метод можно в любой удобный момент. `sendMessage()` тут использовать не нужно.

Для задач IV.2, IV.4

Для начала, вам необходимо отправить информация о маркерах с помощью `sendMessage(msg)`, где `msg` — одна строка, содержащая всю информацию о маркерах, согласно синтаксису в условиях. Обозначу, что это должна быть одна строка, не больше.

То есть, нельзя сначала отправить `sendMessage("20")`, а затем `sendMessage("P1 ... \n P2 ...")`. Нужно отправлять так: `sendMessage("20 \n P1 ... \n P2 ...")` — одной строкой.

Только после того, как вы отправили информацию о маркерах, вы можете вызывать метод `getTask()`, который вернет вам одну строку, согласно синтаксису из условия, то есть, например `"9 \n P3_SW P2_S ..."`.

Далее, при достижении роботом маркера, вы должны сообщить об этом тестирующей системе. Для этого, снова вызываем `sendMessage(msg)`, где `msg` — строка, например, `"P3_SW ОК"`, больше никакой информации быть не должно. Когда робот дошел до следующего маркера, снова вызываем `sendMessage(msg)` с соответствующей информацией, то есть с корректным номером и ориентацией.

Для задачи IV.3

Тоже самое что и для задач IV.2 и IV.4, но теперь роботов два, немного изменятся сообщения, примеры есть в условии.