

## Задания муниципального этапа ВСОШ по информатике по профилю «Искусственный интеллект»

Возрастная группа: 9 класс

Длительность олимпиады: 180 минут

Максимальная оценка: 100 баллов

### ЗАДАНИЕ № 1. Шахматный конь

10 баллов (0 / 2 / 6 / 10)



На стандартной шахматной доске 8×8 в некоторой клетке стоит конь (смотрите рисунок). Несколько других клеток помечены заглавными русскими буквами (каждая буква – ровно один раз). Конь может ходить по обычным шахматным правилам: за один ход он перемещается на 2 клетки по одной координате и на 1 клетку – по другой (всего до 8 возможных направлений).

Для каждой буквы определите минимальное число ходов, за которое конь может добраться из начальной позиции до клетки с этой буквой.

Требуется расположить все буквы в следующем порядке:

1. Сначала – те, до которых конь доходит за наименьшее число ходов.
2. Затем – за следующее по величине число ходов, и так далее.
3. Если до нескольких букв путь занимает одинаковое число ходов, то эти буквы следует расположить в алфавитном порядке.

**ОТВЕТ:** Д В А Б Г

### РЕШЕНИЕ

Будем пошагово определять, за сколько ходов конь может добраться до каждой буквы.

Как можно решать (алгоритм поиска в ширину):

Начало: отмечаем стартовую клетку коня – это 0 ходов.

Первый ход: смотрим все клетки, куда конь может попасть за 1 ход из старта (все возможные ходы буквой «Г»). Эти клетки получают номер 1.

Второй ход: из всех клеток с номером 1 смотрим, куда можно попасть за следующий ход (если клетка ещё не помечена). Эти новые клетки получают номер 2.

Продолжаем так, пока не встретим все буквы.

### ЗАДАНИЕ № 2. Расчёт в лаборатории «НейроБот»

10 баллов (0 / 2,5 / 5 / 7,5 / 10)

В исследовательской лаборатории «НейроБот» запущен эксперимент по распределённому обучению нейросети. В нём участвуют **9 роботов-вычислителей**, и каждый из них обработал **одинаковое количество подзадач**.

В эксперименте используется **ровно 8 учебных задач**. Каждая задача заранее разбита либо на 5, либо на 6, либо на 9 одинаковых подзадач. Все подзадачи были распределены между роботами, и ни одна не осталась невыполненной.

Определите все возможные значения количества подзадач, которые мог обработать каждый робот?

**Инструкция по вводу ответа:** Запишите все возможные значения в порядке возрастания, по одному числу в каждое поле ответа. Количество полей можно увеличивать. Лишние или пропущенные значения будут учитываться при оценке.

**ОТВЕТ:** 5, 6, 7, 8

### РЕШЕНИЕ

Пусть:  $x$  — число задач, разбитых на 5 подзадач,  $y$  — на 6,  $z$  — на 9.

Тогда:  $x + y + z = 8, x, y, z \in \mathbb{Z}_{\geq 0}$ .

Общее число подзадач:  $S = 5x + 6y + 9z$ .

Это число **делится на 9**, так как подзадачи поровну распределены между 9 роботами.

Искомое значение:  $k = S/9$ .

Нужно найти все возможные целые  $k$ , соответствующие допустимым тройкам  $(x, y, z)$ . Находим все верные значения перебором (вручную или программно).

Ниже приведена программа на языке программирования Python, которая находит верные значения перебором всех возможных троек  $(x, y, z)$ .

```
task = 8
for x in range(task + 1):
    for y in range(task + 1):
        for z in range(task + 1):
            if (x + y + z == task) and ((5*x + 6*y + 9*z) % 9 == 0):
                print(x, y, z, (5*x + 6*y + 9*z), (5*x + 6*y + 9*z) // 9)
```

### ЗАДАНИЕ № 3. Парадокс совпадений

10 баллов (0 / 10)

В исследовательской группе по анализу поведенческих данных участвуют  $n$  человек. Предполагается, что день рождения каждого человека равновероятен и независим от других, а год содержит ровно 365 дней (високосные годы не учитываются).

Каково наименьшее значение  $n$ , при котором вероятность того, что хотя бы у двух человек дни рождения совпадают, превышает 0,5?

#### Формат ответа

Выведите одно целое число — искомое значение  $n$ .

**ОТВЕТ:** 23

### РЕШЕНИЕ

Это классический «парадокс дней рождения».

Вероятность отсутствия совпадений:

$$P_{\text{нет}}(n) = \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \dots \cdot \frac{365-n+1}{365}.$$

Нужно найти наименьшее  $n$ , при котором  $1 - P_{\text{нет}}(n) \geq 0,5$ .

#### ЗАДАНИЕ № 4. Эффективность антиспам-фильтра

30 баллов (0 / 5 / 10)

Антиспам-система для каждого входящего письма вычисляет вероятность  $p$  того, что оно является спамом. Для автоматической сортировки используются два порога:

\* Если  $p \leq 0,25$ , письмо помечается как «Не спам» (класс 0) и отправляется в папку «Входящие».

\* Если  $p \geq 0,85$ , письмо однозначно считается спамом (класс 1) и перемещается в папку «Спам».

\* Если  $0,25 < p < 0,85$ , система помечает письмо для ручной проверки администратором (отказ от автоматического принятия решения).

Вам предоставлен датасет в двух эквивалентных файлах, с двумя колонками:  $p$  (предсказанная моделью вероятность) и  $y$  (фактический класс, 0 – не спам, 1 – спам).

Ссылка на данные в формате .ods

Ссылка на данные в формате .xlsx

Определите, сколько писем система обработала абсолютно корректно и автоматически.

В первое поле ввода ответа укажите количество писем (целое число), которые были верно классифицированы как спам (т.е. где  $p \geq 0,85$  и  $y = 1$ ).

Во второе поле ввода ответа укажите количество писем (целое число), которые были верно классифицированы как не спам (т.е. где  $p \leq 0,25$  и  $y = 0$ ).

Сколько подзадач выполнил каждый робот?

**ОТВЕТ:** 30 и 107 соответственно.

#### РЕШЕНИЕ

Для определения количество писем, которые были верно классифицированы как спам можно было ввести в ячейку C2 следующую формулу

=ЕСЛИ (И (A2>=0,85; B2=1) ; 1 ; 0)

и продлить её не весь диапазон данных. Затем посчитать количество единиц в колонке C.

Для определения количество писем, которые были верно классифицированы как не спам можно было ввести в ячейку D2 следующую формулу

=ЕСЛИ (И (A2<=0,25; B2=0) ; 1 ; 0)

и продлить её не весь диапазон данных. Затем посчитать количество единиц в колонке D.

**ЗАДАНИЕ № 5. Поиск симметричных фрагментов**

Ограничения по времени: 1 секунда

Ограничение по памяти: 256 МБ

Максимальная оценка: 30 баллов

В ИИ-системе анализа текста есть модуль, который ищет «**симметричные фрагменты**» - подстроки, начинающиеся и заканчивающиеся **одинаковым символом**. Такие фрагменты могут использоваться для выявления повторяющихся паттернов или ошибок в тексте.

Дана строка, состоящая из заглавных латинских букв. Найдите **самую длинную подстроку**, которая является **симметричной** (начинается и заканчивается одним и тем же символом), и выведите её длину.

**Примечание:** Подстрока определяется как последовательность символов, идущих подряд. Если длина подстроки равна 1, она считается симметричной.

**Формат входных данных**

Одна строка – последовательность заглавных латинских букв.

Длина строки  $1 \leq s \leq 10^5$ .

**Формат выходных данных**

Одно целое число – максимальная длина симметричной подстроки.

**Система оценивания**

Программа, правильно работающая и дающая верные ответы на все тесты, будет оцениваться в 30 баллов. Каждый тест оценивается независимо, возможны частичные решения.

**Пояснение к примеру**

Подстрока **АВАСА** (с 1-го по 5-й символ) начинается и заканчивается на **А**, её длина – 5. Большей симметричной подстроки нет.

Ввод	Вывод
ABACAD	5

**РЕШЕНИЕ**

Ниже приведено одно из возможных решений задачи на языке программирования Python.

```
m = {}
ans = 1
for i, c in enumerate(input()):
    if c in m:
        ans = max(ans, i - m[c] + 1)
    else:
        m[c] = i

print(ans)
```

### ЗАДАНИЕ № 6. Оптимальный порог классификации

Ограничения по времени: 1 секунда

Ограничение по памяти: 512 МБ

Максимальная оценка: 30 баллов

На числовой прямой расположены точки двух классов:

- **Синие** (класс 0) — это объекты, которые модель должна считать «низкими».
- **Красные** (класс 1) — «высокие».

Классификатор не видит цвета, но может использовать **порог  $T$** :

- если координата точки  $< T \rightarrow$  считает её **синей**,
- если координата  $\geq T \rightarrow$  считает её **красной**.

Для оценки качества используются две ошибки:

- **Ошибка I рода**: доля синих точек, ошибочно отнесённых к красным

$$= \frac{\text{число синих точек} \geq T}{\text{всего синих}};$$

- **Ошибка II рода**: доля красных точек, ошибочно отнесённых к синим

$$= \frac{\text{число красных точек} < T}{\text{всего красных}}.$$

Ваша задача – найти **целое значение порога  $T$** , при котором **сумма этих двух ошибок минимальна**. Если таких значений несколько, выберите то, для которого значение  $T^2$  минимально, если и таких значений несколько, выберите минимальное неотрицательное.

#### Формат входных данных

- Первая строка: целое число  $m$  — количество синих точек ( $1 \leq m \leq 1000$ ), затем  $m$  целых чисел – их координаты.
- Вторая строка: целое число  $k$  — количество красных точек ( $1 \leq k \leq 1000$ ), затем  $k$  целых чисел – их координаты.

Все координаты — целые, по модулю не превышают  $10^3$ .

#### Формат выходных данных

Выведите одно целое число – искомое значение порога  $T$ .

#### Система оценки

Программа, правильно работающая и дающая верные ответы на все тесты, будет оцениваться в 30 баллов. Каждый тест оценивается независимо, возможны частичные решения.

#### Пояснение к примеру

При  $T = 4$ :

- Синие точки ( $\geq 4$ ): 7, 10, 11, 15  $\rightarrow$  4 из 6  $\rightarrow$  ошибка I рода = 4/6
- Красные точки ( $< 4$ ): 0 из 4  $\rightarrow$  ошибка II рода = 0/4
- Сумма = 4/6 + 0/4 = 2/3 + 0  $\approx 0.667$

Проверяя все разумные  $T$ , можно убедиться, что при  $T = 4$  сумма ошибок минимальна, и среди всех таких  $T$  это наименьшее.

Ввод	Вывод
6 1 3 7 10 11 15 4 4 6 8 13	4

**РЕШЕНИЕ**

Ниже приведено одно из возможных решений задачи на языке программирования Python.

```

m, *am = map(int, input().split())
k, *ak = map(int, input().split())
t_begin = min(min(am), min(ak))
t_end = max(max(am), max(ak))

t_min = t_end
error = 2
for i in range(t_begin-10, t_end + 10):
    error_one = len([x for x in am if x >= i]) / m
    error_two = len([x for x in ak if x < i]) / k
    if error_one + error_two < error:
        error = error_one + error_two
        t_min = i
    elif error_one + error_two == error and (abs(i) <= abs(t_min)):
        t_min = i

print(t_min)

```