

**Всероссийская олимпиада школьников
по информатике 2025-2026
Профиль “Искусственный интеллект”**

Районный тур, 7-11 класс
Санкт-Петербург, 1 декабря 2025 года

РАЗБОР ЗАДАЧ

Задача А - Род слов

Слово женского рода, если оно заканчивается на:

- а
- ?d, ?z, где ? – гласная
- ion

Гласные – а, е, о, і, и

Определить род слова

casa, puerta, libertad,
mesa, lluvia

libro, amico, treno,
amore, sole

chapeau, livre, jardin,
cafe, ami

Задача А - код

```
word = input()
```

```
vowels = set("aeiou")
```

```
fem = (  
    word.endswith("a") or  
    (len(word) >= 2 and word[-1] in "dz" and word[-2] in vowels) or  
    word.endswith("ion")  
)
```

```
print("f" if fem else "m")
```

Задача В - Род слов - обучаем ИИ

Слово женского рода, если оно заканчивается на:

- a
- ?d, ?z, где ? – гласная
- ion

Гласные – a, e, o, i, u

Сгенерировать 120 слов с разметкой

casa, puerta, libertad,
mesa, lluvia

libro, amico, treno,
amore, sole

chapeau, livre, jardin,
cafe, ami

Задача В - Род слов - обучаем ИИ

Женский род

- *а - 20 слов
- ?d, ?z, где ? - гласная - по 2 для каждой гласной а, е, о, і, у, и каждой согласной d и z, всего 20
- іоп - 20 слов

Мужской род

- 60 слов

Задача В - Женский род, первый класс

Женский род

- *a - 20 слов

```
cons = "bcdfghjklmnpqrstvwxyz"
```

```
for i in range(20):  
    female_words.append(cons[i] + 'a')
```

Задача В - Женский род, второй класс

Женский род

- ?d, ?z, где ? – гласная – по 2 для каждой гласной а, е, о, і, u, и каждой согласной d и z, всего 20

```
vowels = "aeiou"
```

```
for prefix in ['b', 'c']:
    for v in vowels:
        for suffix in ['d', 'z']
            female_words.append(prefix + v + suffix)
```

Задача В - Женский род, третий

Женский род

- *ion - 20 слов

```
cons = "bcdfghjklmnpqrstvwxyz"
```

```
for i in range(20):  
    female_words.append(cons[i] + 'ion')
```

Задача В - Мужской род

Мужской род

- 60 слов

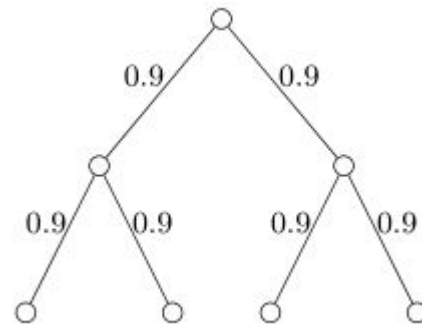
```
cons = "bcdfghjklmnpqrstvwxyz"
```

```
for i in range(20):  
    for suffix in ['o', 'e', 'u']  
        male_words.append(cons[i] + suffix)
```

Задача С - Обновление базы данных

Дано дерево, с вероятностью 0.9
каждое ребро целое, с вероятностью
0.1 оно повреждается

Найти среднее число листьев, у
которых будет путь до корня для
высоты 2 и 20

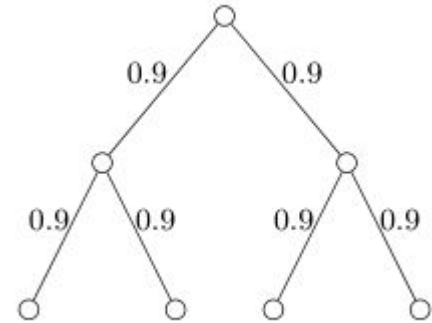


Задача С - Обновление базы данных - решение

Вероятность, что конкретный лист имеет путь до корня равна 0.9^h

Всего листьев 2^h

Матожидание линейно, значит ответ равен $0.9^h * 2^h = 1.8^h$



Задача С - Обновление базы данных - решение

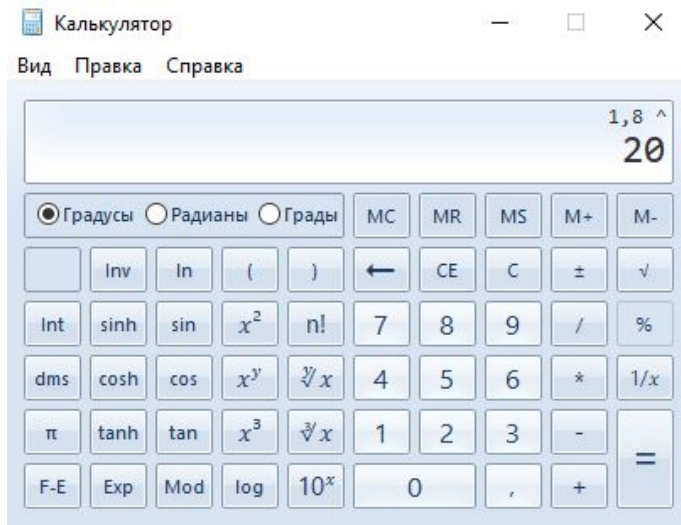
Вероятность, что конкретный лист имеет путь до корня равна 0.9^h

Всего листьев 2^h

Матожидание линейно, значит ответ равен $0.9^h * 2^h = 1.8^h$

$h = 2, \text{ ans} = 3.24$

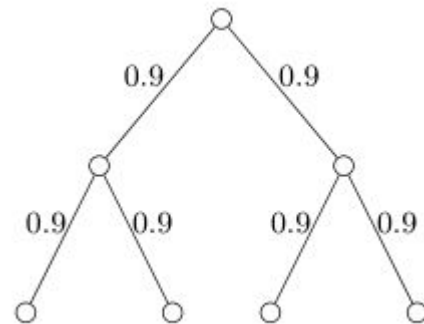
$h = 20, \text{ ans} = 127482.36216396085$



Задача D - Обновление базы данных, часть 2

Дано дерево, с вероятностью 0.9 каждое ребро целое, с вероятностью 0.1 оно повреждается

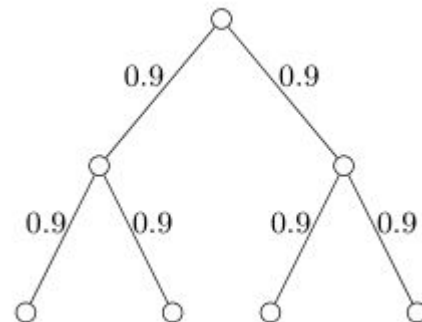
Найти вероятность, что от листа до корня есть путь хотя бы у половины листьев для $h = 2$ и $h = 20$



Задача D - Обновление базы данных, часть 2

Для $h = 2$ посчитаем, перебрав 2^6 вариантов, какие ребра остались целыми.

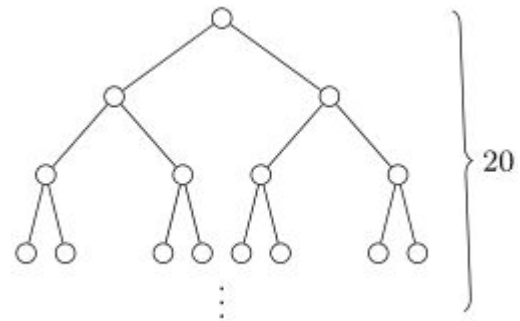
Вероятность оказывается равна 0.952803



Задача D - Обновление базы данных, часть 2

Для $h = 20$ так не работает.

Есть два подхода: математический и программистский



Задача D - Обновление базы данных, часть 2

Математический: среднее значение мы посчитали в C, равно примерно 120 тыс

Надо, чтобы было хотя бы $2^{20}/2$, то есть примерно 520 тыс

Вероятность, что случайная величина так сильно отклонится от матожидания пренебрежимо мала, по неравенству Чернова порядка 10^{-56000}

Величины у нас не независимые, но все равно ясно, что значение мало

- Верхний хвост ($\delta \in (0, 1]$):
 $\mathbb{P}(X \geq (1 + \delta)\mu) \leq e^{-\delta^2\mu/3}.$

Задача D - Обновление базы данных, часть 2

Программистский: напишем ДП, которое считает вероятность $prob[k][i]$, что на уровне k выживут i вершин

ДП работает за квадрат числа вершин на высоту, для $h=20$ не запустить, но видно что начинает очень быстро убывать

Нам нужна точность 10^{-6}

Вывод: $h = 20, ans = 0$

```
1 0.99
2 0.9528030000000001
3 0.8851513094923502
4 0.7932999416788977
5 0.7020116197514759
6 0.6146278059591291
7 0.5053610263033509
8 0.37102598462433767
9 0.22633511908726914
10 0.09865181095059575
11 0.02255444756786245
12 0.0014839705626535499
13 8.377473231221169e-06
```

Задача E - 1-НОТ кодирование

Дан набор объектов

Для i -го объекта 1-НОТ код это вектор, где все нули и на i -й позиции единица

Вывести 1-НОТ кодирование

стандартный ввод	стандартный вывод
4	1 0 0 0
cat	0 0 0 1
dog	0 0 0 1
mouse	0 0 1 0
bird	0 1 0 0
5	
cat	
bird	
bird	
mouse	
dog	

Задача E - 1-НОТ кодирование - решение

Задача на техническую реализацию

Запомним элементы в массиве или словаре

Для каждого запроса выведем необходимый вектор

(но задача знакомит нас с важным понятием в машинном обучении, простим авторам такую тривиальность)

Задача E - код

```
n = int(input())
d = {}
for i in range(n):
    s = input()
    d[s] = i
q = int(input())
for i in range(q):
    s = input()
    for i in range(d[s]):
        print("0 ", end="")
    print("1 ", end="")
    for i in range(n - d[s] - 1):
        print("0 ", end="")
    print()
```

Задача F - квадратный корень

Найти квадратный корень из матрицы, где один из элементов равен нулю

$$A = \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}$$

$$C = AB = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$$

Задача F - решение

Будем искать матрицу в таком же виде

Подставляя в формулу, видим, что

$$x_{11} * x_{11} = a$$

$$x_{11} * x_{12} + x_{12} * x_{22} = b$$

$$x_{22} * x_{22} = c$$

Получается $x_{11} = \sqrt{a}$, $x_{22} = \sqrt{c}$

$$x_{21} = 0, \quad x_{12} = b / (x_{11} + x_{22})$$

Задача F - код

```
from math import sqrt
```

```
A, B, C = map(float, input().split())
```

```
a = sqrt(A)
```

```
c = sqrt(C)
```

```
b = B / (a + c)
```

```
print(a, b, 0, c)
```

Задача G - ограничение скорости

Требуется отслеживать ограничения скорости, устанавливаемые знаками, началом и концом населенного пункта и автомагистрали



Задача G - ограничение скорости - код

```
n = int(input())
zone = "city" # city, nocity, highway
temp_limit = None
default_speeds = {"city": 60, "nocity": 90, "highway": 110}
speed = 60
prev_zone = "city"
```

Задача G - ограничение скорости - код

```
for i in range(n):
    event = input().split()
    type_event = event[0]
    if type_event == "city":
        zone = "city"
    elif type_event == "nocity":
        zone = "nocity"
    elif type_event == "highway":
        prev_zone = zone
        zone = "highway"
    elif type_event == "nohighway":
        zone = prev_zone
    if type_event == "limit":
        speed = int(event[1])
    else:
        speed = default_speeds[zone]
    print(speed)
```

Задача H - высокий спрос

Есть n пассажиров, i -й пассажир готов заплатить $A_i + B$

Есть m таксистов, j -й таксист готов возить за $C_j + D$

Какую цену установить, чтобы перевезти максимальное число пассажиров? Какую максимальную цену можно установить для такого числа пассажиров?

Задача N - решение

Пусть установили цену x

Тогда поедут пассажиры, для которых

$Ai + B \geq x$, то есть $i \geq (x - B) / A$, то есть $i \geq \text{ceil}((x - B) / A)$

таких $n - \text{ceil}((x - B) / A)$

И повезут водители, для которых

$Cj + D \leq x$, то есть $j \leq (x - D) / C$, то есть $j \leq \text{floor}((x - D) / C)$,

таких $\text{floor}((x - D) / C) + 1$

Задача N - решение

Значит надо найти такое x , чтобы

$$n - \text{ceil}((x-B)/A) \leq \text{floor}((x-D)/C)+1$$

Из таких максимальное

Это можно сделать бинарным поиском, либо аккуратно раскрыв формулы

Задача I - Анализ оценок

Задан cvs-файл

Требуется загрузить данные, посчитать среднее арифметическое оценок тех, у кого три или больше оценок

Среди таких найти классы, где среднее значение этой величины у мальчиков и у девочек максимально различается в ту и другую сторону

Задача I - решение

Воспользуемся модулем csv в Python

```
# --- Read students.csv ---
students = {} # student_id -> (gender, class)
with open('students.csv', encoding='utf-8') as f:
    reader = csv.reader(f, delimiter=';')
    next(reader, None) # skip header
    for row in reader:
        sid_str, surname, name, gender, cls = row
        sid = int(sid_str)
        students[sid] = (gender, cls)
```

Задача I - решение

Воспользуемся модулем csv в Python

```
# --- Read grades.csv ---
with open('grades.csv', encoding='utf-8') as f:
    reader = csv.reader(f, delimiter=';')
    next(reader, None) # skip header
    for row in reader:
        grade_id, sid_str, subject_id, grade_str = row
        sid = int(sid_str)
        grade = int(grade_str)
        if sid not in grades:
            grades[sid] = []
        grades[sid].append(grade)
```

Задача I - решение

Теперь, когда данные считаны, обработать их – дело техники
Считаем средние оценки

```
student_avg = {} # sid -> avg_grade
for sid, gr_list in grades.items():
    if len(gr_list) >= 3:
        student_avg[sid] = sum(gr_list) / len(gr_list)
```

Задача I - решение

Усредняем по классам и полам

```
class_stats = defaultdict(lambda: {"f": [], "m": []})
for sid, avg in student_avg.items():
    gender, cls = students[sid]
    if gender in ("f", "m"):
        class_stats[cls][gender].append(avg)
```

Задача I - решение

Вычисляем то, что у нас спрашивали

```
diffs = []
for cls, d in class_stats.items():
    if d["f"] and d["m"]:
        Fi = sum(d["f"]) / len(d["f"])
        Mi = sum(d["m"]) / len(d["m"])
        diffs.append(Fi - Mi)
Dmax = max(diffs)
Dmin = min(diffs)

print(Dmax, Dmin)
```

Спасибо за внимание

spbmunicipal@gmail.com