Максимальное количество баллов за олимпиаду — 72
Максимальный балл за задание — 12

Задание 1.

Представим, что число меняется по следующему алгоритму: если число чётно, его делят на 2, а если нечётно — вычитают 1. Из числа N после выполнения четырёх шагов алгоритма получилось число 12. Определите наименьшее возможное значение N.

Ответ. 51.

Критерии оценивания. Верный ответ — 12 баллов; неверный или отсутствующий ответ — 0 баллов.

Решение.

Число 12 могло быть получено из 13 или 24. Число 13 — только из 26, а 26 — из чисел 27 и 52, но ни одно из них не получить из числа, меньшего 51, за одну операцию. Число 24 можно получить из чисел 48 или 25, но 48 нельзя получить за две операции из числа, меньшего 51. Перед числом 25 могло быть только 50, поэтому перед выполнением шагов алгоритма минимальное возможное число — 51.

Задание 2.

Модель искусственного интеллекта различает цифры 0 и 1 по следующему алгоритму: если линия, которой нарисована цифра, ограничивает некоторую часть плоскости, цифра определяется как 0, иначе — как 1. В частности, цифру 0 она определяет как 0, а цифру 1 — как 1. Полученная последовательность цифр интерпретируется как число в двоичной системе счисления. Какое число модель определит по записи 293676? Ответ запишите в десятичной системе счисления.

Ответ. 42.

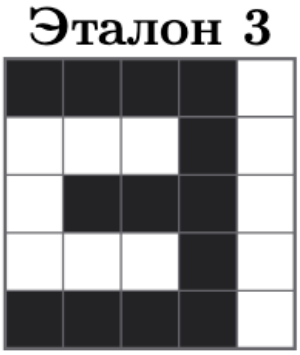
Критерии оценивания. Верный ответ — 12 баллов; неверный или отсутствующий ответ — 0 баллов.

Решение.

Следуя условию задачи, цифры 9 и 6 будут посчитаны как нули, остальные — как единицы, в двоичной системе получится число 101010. Переводим в десятичную систему счисления: $2 + 8 + 32 = 42$.

Задание 3.

Модель искусственного интеллекта различает пиксельные цифры 3 и 8. Для каждой цифры у модели есть эталонное изображение:





Расстоянием между двумя изображениями называется число клеток, цвет которых на этих изображениях различается. Модель сравнивает новое изображение с обоими эталонами и выбирает цифру с меньшим расстоянием. Если расстояния равны, модель выбирает цифру 8.

Модели предложили изображение ниже. Строки и столбцы пронумерованы сверху вниз и слева направо соответственно.



За одно изменение можно поменять цвет одной клетки: белый на чёрный или чёрный на белый.

Найдите расстояние от данного изображения до эталона 3. Найдите расстояние от данного изображения до эталона 8. Какую цифру выберет модель? Сколько клеток данного изображения обладают следующим свойством: изменение цвета этой клетки изменит ответ модели с 3 на 8? Может ли изменение клетки, цвет которой совпадает в двух эталонах, изменить ответ модели? Правило выбора при равенстве расстояний изменили: теперь в случае равенства модель выбирает цифру 3. Какое минимальное число клеток данного изображения нужно изменить, чтобы модель выбрала цифру 8?

Варианты ответа:

1. Да

2. Нет

Ответ. Расстояния 4 и 6; выбрана цифра 3; 5 клеток; нет (вариант 2); минимум 2 изменения.

Критерии оценивания. За правильные ответы по порядку шести вопросов: 1; 1; 1; 4; 2; 3 балла соответственно. Баллы за пункты суммируются, максимум — 12. За неверный или отсутствующий ответ на отдельно оцениваемый пункт — 0 баллов, если частичное оценивание не указано отдельно.

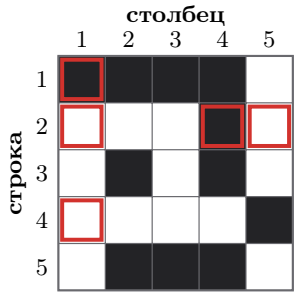
Решение.

1. Данное изображение отличается от эталона 3 в 4 клетках, а от эталона 8 — в 6 клетках. Поскольку $4 < 6$, модель выбирает цифру 3.

   Ответ: расстояния 4 и 6; модель выберет 3.

2. Чтобы ответ изменился на 8, нужно увеличить расстояние до эталона 3 и одновременно уменьшить расстояние до эталона 8. Для этого цвет клетки данного изображения должен совпадать с эталоном 3 и отличаться от эталона 8.

   Таких клеток пять:



   После изменения любой из них оба расстояния станут равны 5, и по правилу из условия модель выберет 8.

   Это клетки $(1, 1)$, $(2, 1)$, $(2, 4)$, $(2, 5)$ и $(4, 1)$.

   Ответ: 5 клеток.

3. Нет. Если цвет клетки в эталонах совпадает, то после изменения этой клетки оба расстояния либо увеличатся на 1, либо уменьшатся на 1. Поэтому разность расстояний не изменится и тот же эталон останется ближе.

   Ответ: нет.

4. Изначально расстояние до эталона 3 равно 4, а до эталона 8 равно 6, поэтому разность $d_8 - d_3$ равна 2. При изменении одной клетки каждое из расстояний изменяется ровно на 1: совпадение цвета с эталоном превращается в несовпадение или наоборот. Если цвет этой клетки в эталонах одинаков, оба расстояния изменятся в одну сторону и их разность не изменится. Если цвета различаются, расстояния изменятся в разные стороны и их разность изменится ровно на 2 по модулю. Следовательно, за одно изменение разность $d_8 - d_3$ можно уменьшить не более чем на 2. Поэтому после одного изменения в лучшем случае получится равенство $d_3 = d_8 = 5$, при котором по новому правилу модель всё равно выберет цифру 3.

   Двух изменений достаточно. Например, изменим клетки $(1, 1)$ и $(2, 1)$. Обе клетки до изменения совпадают с эталоном 3 и отличаются от эталона 8, поэтому каждое изменение увеличивает d_3 на 1 и уменьшает d_8 на 1. Получаем

$$d_3 = 4 + 2 = 6, \qquad d_8 = 6 - 2 = 4.$$

   Теперь модель выберет цифру 8.

   Ответ: 2 клетки; например, $(1, 1)$ и $(2, 1)$.

Задание 4.

Приёмная комиссия обрабатывает заявки. Каждая заявка описывается четырьмя признаками со значениями «да» или «нет»:

А — есть подпись;

Б — оплачен взнос;

В — заявка подана онлайн;

Г — приложена фотография.

Заявку принимают ровно тогда, когда одновременно есть подпись и оплачен взнос. Признаки В и Г на верный ответ не влияют. Всего существует 16 различных заявок — по одной для каждого набора значений признаков.

Сравниваются две модели искусственного интеллекта. Модель П отвечает «принять» ровно тогда, когда заявка подана онлайн. Модель Р отвечает «принять» ровно тогда, когда есть подпись.

Сколько заявок должно быть принято по правилам приёмной комиссии? Сколько заявок примет модель П? Сколько заявок примет модель Р?

Разделим все заявки на четыре группы. Определите размеры этих групп в каждом из случаев.

Обе модели ответили верно:

Только модель П ответила верно:

Только модель Р ответила верно:

Обе модели ошиблись:

Качество моделей сравнивают на валидационной выборке — некотором наборе различных заявок. Какое наибольшее число заявок может содержать валидационная выборка, на которой модель П ошибается реже, чем модель Р?

Ответ. По порядку вопросов: 4; 8; 8; 6; 2; 6; 2; 11.

Критерии оценивания. По 1 баллу за каждый из первых семи верных ответов; 5 баллов за последний ответ. Баллы суммируются, максимум — 12. За неверный или отсутствующий ответ на отдельный вопрос — 0 баллов.

Решение.

1. Для принятия нужны значения «да» для признаков А и Б, а два остальных признака выбираются произвольно. Поэтому принять нужно $2^2 = 4$ заявки. Каждая модель учитывает только один признак, поэтому каждая из них принимает половину из 16 заявок, то есть 8.

   Ответ: 4, 8, 8.

2. Модель Р ошибается только на заявках с подписью, но без взноса: их принимать нельзя, а Р их принимает. Таких заявок 4, остальные 12 модель Р определяет верно.

   Решения модели П верны для тех из 4 подлежащих принятию заявок, которые поданы онлайн, — их 2, и для тех из 12 остальных заявок, которые поданы не онлайн, — их 6.

   Из 4 ошибок модели Р 2 допущены в онлайн-заявках, на них П тоже ошибается; на остальных 2 модель П верна. Значит, только П справляется с 2 заявками, обе модели ошибаются на 2 заявках. Обе модели справляются с $8-2=6$ заявками, а только Р верно отвечает на $12-6=6$ заявок.

   Ответ: 6, 2, 6, 2.

3. Возьмём все 8 заявок, которые модели оценивают одинаково. В этом случае обе модели имеют одинаковое число ошибок. Добавим все 2 заявки, верно оценённые только моделью П, и не более одной заявки, на которой верна только модель Р. Получится выборка из

   $$8+2+1=11$$

   заявок, где модель П ошибается реже.

   Больше заявок взять нельзя. При обработке заявок, на которые модели отвечают различно, модель П должна получить преимущество. Таких заявок, на которые верно отвечает только П, всего 2, поэтому заявок противоположного типа можно взять не больше одной. К ним можно добавить не больше 8 заявок, где модели отвечают одинаково.

   Ответ: 11.

Задание 5.

Модель должна обработать 9 запросов, длины которых равны 34, 8, 17, 11, 46, 21, 4, 41 и 20 токенам соответственно.

Вы можете произвольным образом разбить эти запросы на 3 батча (группы) по 3 запроса в каждом так, чтобы каждый запрос попал ровно в один из батчей. Вычислительные затраты на обработку одного батча равны длине самого длинного запроса в этом батче.

Найдите минимальные суммарные вычислительные затраты на обработку всех запросов при оптимальном разбиении на батчи.

Ответ. 78.

Критерии оценивания. Верный ответ — 12 баллов; неверный или отсутствующий ответ — 0 баллов.

Решение.

Упорядочим длины по возрастанию:

$$4,\ 8,\ \mathbf{11},\ 17,\ 20,\ \mathbf{21},\ 34,\ 41,\ \mathbf{46}.$$

Оценка. Пусть $M_1 \geqslant M_2 \geqslant M_3$ — стоимости батчей, то есть длины самых длинных запросов в них.

В батче со стоимостью M_3 есть ещё два запроса, и оба не длиннее M_3. Значит, M_3 не меньше третьего числа списка: $M_3 \geqslant 11$.

Запрос длины M_2 не короче трёх запросов из батча с M_3 и двух запросов из своего батча — в сумме пять. Значит, M_2 не меньше шестого числа списка: $M_2 \geqslant 21$.

Наконец, M_1 — длина самого длинного запроса вообще, то есть $M_1 = 46$.

Итого затраты не меньше $46 + 21 + 11 = 78$.

Пример. Батчи $\{46, 41, 34\}$, $\{21, 20, 17\}$ и $\{11, 8, 4\}$ дают ровно $46 + 21 + 11 = 78$.

Задание 6.

Кластеризация без ML

Ограничение по времени: 3 секунды

Ограничение по памяти: 256 мегабайт

Кластеризация — это разбиение объектов на группы, кластеры, так, чтобы похожие объекты попадали в одну группу, а непохожие — в разные. Обычно кластеры ищет алгоритм, но в этой задаче разбиение складывается само, и его остаётся только измерить.

Биологи выращивают колонию бактерий в стопке из k одинаковых слоёв питательной среды. Каждый слой — прямоугольник из n строк и m столбцов клеток, и в каждой клетке живёт ровно одна бактерия: синяя, красная или зелёная.

Все слои устроены одинаково: в клетке (i, j) любого слоя живёт бактерия одного и того же цвета.

Две бактерии соседние, если их клетки имеют общую грань: это либо соседи по стороне в одном слое, либо клетки с одинаковыми координатами в соседних слоях. Соседние бактерии срастаются в двух случаях:

обе красные или обе зелёные и лежат в одном слое;

обе синие и лежат в соседних слоях.

Других срастаний нет: в частности, две синие бактерии одного слоя не срастаются никогда. Кластер — это множество бактерий, связанных срастаниями, возможно, через цепочку других бактерий.

Граница кластеров состоит из единичных граней клеток, каждая из которых либо разделяет бактерии из разных кластеров, либо отделяет бактерию от внешнего пространства. Помогите биологам посчитать, из скольких граней состоит граница кластеров.

Формат входных данных

В первой строке даны три целых числа n, m и k — количество строк и столбцов в слое и количество слоёв ($1 \leqslant n$, $m \leqslant 2 \cdot 10^5$; $n \cdot m \leqslant 2 \cdot 10^5$; $1 \leqslant k \leqslant 1\,000$).

В каждой из следующих n строк дано по m символов, каждый из которых равен B, R или G. j-й символ i-й из этих строк задаёт цвет бактерии в клетке (i, j) каждого слоя: B — синяя, R — красная, G — зелёная.

Формат выходных данных

Выведите одно целое число — количество граней в границе кластеров.

Пример 1

Ввод	**Вывод**
2 3 2 RRG RBG	45

Пример 2

Ввод	**Вывод**
1 1 1 R	6

Пример 3

Ввод	**Вывод**
1 2 1 BB	11

Система оценивания

Тесты из условия не оцениваются. Помимо них в задаче 12 тестов, каждый из которых оценивается в 1 балл.

Замечание

В первом примере колония состоит из двух слоёв, каждый из них такой:

$R \quad R \quad G$
$R \quad B \quad G$

Сначала посчитаем грани внутри каждого слоя. Наружу по периметру выходят $2 \cdot (2 + 3) = 10$ граней. Внутри слоя границей являются 4 грани: между R и G в первой строке, между R и B и между B и G во второй строке и между R и B в среднем столбце. Две клетки R в первом столбце срослись, и грани между ними в границе нет. Получается 14 граней на слой, 28 на два слоя.

Теперь грани между слоями. Снизу и сверху колонии наружу выходят по 6 граней. Между слоями лежат 6 граней, и границей являются 5 из них: только в клетке B синие бактерии срослись в вертикальную нить. Всего $6 + 6 + 5 = 17$ граней, а ответ равен $28 + 17 = 45$.

Ответ. $2nm + 2(n + m)k + kE + (k - 1)C$, где E — число пар соседних по стороне клеток одного слоя, которые не срастаются (цвета различны или обе клетки синие), а C — число несиних клеток одного слоя.

Критерии оценивания. Программа выводит точное число граней; общая граница разных кластеров учитывается один раз. Всего 12 оцениваемых тестов. Тесты из условия не оцениваются. Каждый из остальных тестов оценивается независимо в 1 балл; непройденный тест — 0 баллов. Максимум — 12 баллов.

Решение.

Рассмотрим отдельно несколько типов граней.

Сначала посчитаем внешние грани всей конструкции.

Сверху и снизу находится по $n \cdot m$ граней, поэтому они дают $2nm$.

Кроме того, по бокам каждого слоя находится $2(n+m)$ граней. Всего слоёв k, поэтому боковые грани дают $2(n+m)k$.

Теперь рассмотрим грани между соседними клетками внутри одного слоя.

Если две соседние клетки имеют одинаковый цвет и этот цвет равен `R` или `G`, то бактерии срастаются и граница между ними не учитывается.

Во всех остальных случаях грань между этими клетками входит в границу. Поскольку все k слоёв одинаковые, каждая такая пара клеток даёт сразу k граней.

Остаётся посчитать грани между соседними слоями.

В клетке цвета `B` бактерии из соседних слоёв срастаются, поэтому границ между ними нет. В клетках цветов `R` и `G` срастания между слоями не происходит. Для каждой такой клетки имеется $k - 1$ граней между соседними слоями.

Таким образом, достаточно один раз пройти по всей таблице и по всем парам соседних клеток.

Сложность: $O(nm)$ по времени и $O(nm)$ по памяти.

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

field = []

for i in range(n):
    row = input()
    field.append(row)

answer = 0

answer += 2 * n * m
answer += 2 * (n + m) * k

for i in range(n):
    for j in range(m - 1):
        left = field[i][j]
        right = field[i][j + 1]

        merged = False

        if left == right:
            if left == "R" or left == "G":
                merged = True

        if not merged:
            answer += k

for i in range(n - 1):
    for j in range(m):
        upper = field[i][j]
        lower = field[i + 1][j]

        merged = False

        if upper == lower:
            if upper == "R" or upper == "G":
                merged = True

        if not merged:
            answer += k

for i in range(n):
    for j in range(m):
        color = field[i][j]

        if color != "B":
            answer += k - 1

print(answer)
```

Альтернативное решение

Можно начать с суммы площадей всех отдельных клеток.

Всего клеток в конструкции nmk, у каждой клетки 6 граней. Поэтому сначала получаем

$$6nmk.$$

Однако общая грань двух соседних клеток при таком подсчёте учитывается дважды.

Если две соседние бактерии не срослись, эта грань является границей между двумя кластерами и должна учитываться ровно один раз. Поэтому для каждой пары соседних клеток нужно вычесть 1.

Если же две соседние бактерии срослись, общая грань вообще не входит в границу. В этом случае нужно вычесть ещё

1.

Таким образом,

$$\text{ответ} = 6nmk - \text{число пар соседних клеток} - \text{число срастаний}.$$

Посчитаем число пар соседних клеток.

Внутри одного слоя горизонтальных соседств

$$n(m-1),$$

а вертикальных

$$(n-1)m.$$

Таких слоёв k.

Между соседними слоями находится

$$nm(k-1)$$

пар клеток.

Поэтому общее число соседств равно

$$k\big(n(m-1)+(n-1)m\big)+nm(k-1).$$

Остаётся посчитать число срастаний.

Внутри слоя срастаются соседние клетки одинакового цвета `R` или `G`. Каждое такое соседство повторяется во всех k слоях.

Между слоями срастаются только клетки цвета `B`. Для каждой синей клетки имеется $k-1$ таких срастаний.

Сложность: $O(nm)$ по времени и $O(nm)$ по памяти.

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

field = []

for i in range(n):
    row = input()
    field.append(row)

all_neighbors = 0

all_neighbors += n * (m - 1) * k
all_neighbors += (n - 1) * m * k
all_neighbors += n * m * (k - 1)

merged = 0

for i in range(n):
    for j in range(m - 1):
        left = field[i][j]
        right = field[i][j + 1]

        if left == right:
            if left == "R" or left == "G":
                merged += k

for i in range(n - 1):
    for j in range(m):
        upper = field[i][j]
        lower = field[i + 1][j]

        if upper == lower:
            if upper == "R" or upper == "G":
                merged += k

for i in range(n):
    for j in range(m):
```

```
        if field[i][j] == "B":
            merged += k - 1

answer = 6 * n * m * k
answer -= all_neighbors
answer -= merged

print(answer)
```

