ВСОШ
ВСЕРОССИЙСКАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Муниципальный этап Всероссийской олимпиады школьников 2022/2023 учебный год

Предмет	Класс	Дата	Время начала	Время окончания
Информатика и ИКТ	9-11	9.12.2022		

Рекомендации к решению задач

Задача 1. Почти анаграмма

Подсчитаем, сколько раз встречается каждая буква латинского алфавита в первом и втором слове по отдельности. Для этого создадим два массива счетчиков размером в 26 элементов. Счетчик для буквы `ch` хранится в элементе массива с индексом `ord(ch)-ord('a')`. Если счетчики соответствующих букв в двух словах одинаковы, то слова являются анаграммами. Если счетчики равны для всех букв, кроме одной, и для этой буквы счетчики отличаются ровно на единицу, то ответ YES +/YES -. В остальных случаях ответ — NO.

Задача 2. Прогулка

Так как пешеход может совершить только один прыжок, нас интересуют минимальный и максимальный номера плиток с ямами. Если ям нет, то ($M == 0$) и ответ равен 1. В остальных случаях находим минимальный (*min*) и максимальный (*max*) номера плиток с ямами. Количество способов равно количеству плиток, с которых можно совершить прыжок сразу через все ямы. Из плитки с номером i можно совершить прыжок, если $i < min$, $i + d + 1 > max$ и $i + d + 1 \le N$. Проверять плитки можно в цикле, или вывести формулы для различных случаев:

1. после плитки с номером *max* достаточно места, чтобы не перепрыгнуть за последнюю плитку;
2. перед плиткой с номером *min* достаточно места и можно найти плитку, с которой можно сделать прыжок на плитку с номером $max+1$;
3. случаи 1,2 не выполняются, но длины прыжка хватает, чтобы перепрыгнуть все ямы.

Задача 3. Ахиллес, черепаха и муха

Известно, что расстояние — произведение скорости на время. Для того, чтобы узнать расстояние, которое пролетит муха, нужно узнать время полета.
Для этого выясним, сколько времени Ахиллес будет преследовать черепаху. Заметим, что если расстояние между Ахиллесом и черепахой равно 1, то Ахиллесу бежать не нужно.
Если расстояние меньше или равно 2, то нужно бежать 1 минуту, если меньше или равно 4, то нужно бежать 2 минуты и т.д.
Обобщая, можно получить следующее "разбиение" возможных расстояний на время бега: (1, 2], (2, 4], ... $(2^{k-1}, 2^k]$, ...
Если изначальное расстояние попадает в полуинтервал $(2^{k-1}, 2^k]$, то Ахиллесу необходимо k минут.
Поэтому чтобы найти время полета мухи, нужно найти подходящий отрезок.
Найдя время, нужно вспомнить, что размерность скорости мухи — м/с, а время дано в минутах, поэтому нужно время перевести в секунды и посчитать итоговый ответ.
Пример решения на языке Python 3.7:

```
d, v = map(int, input().split())
v *= 60
c = 1
t = 0
while c < d:
  c *= 2
  t += 1

print(v * t)
```

Предмет	Класс	Дата	Время начала	Время окончания
Информатика и ИКТ	9-11	9.12.2022		

Задача 4. Защитные башни

Для решения задачи необходимо разделить руины (точки) на 2 набора: те, что находятся по одну сторону от реки (прямой), и по другую сторону от нее. Сделать это можно, вычисляя значение выражения $a{\cdot}x + b{\cdot}y + c$. Если значение — отрицательное число, то будем считать, что точка (x, y) находится на первой стороне от прямой. Если значение выражения — положительное число, точка (x, y) находится на второй стороне от
прямой.
Вариант неэффективного по времени решения.

- Попарно переберем все точки из набора i (i=1,2).
- Для пары точек (x_1, y_1) и (x_2, y_2) проверим можно ли образовать прямоугольник, в котором эти точки будут диагонально противоположны. Две точки образуют такой прямоугольник, если в наборе i существуют точки (x_1, y_2) и (x_2, y_1).
- Для проверки существования точек (x_1, y_2) и (x_2, y_1) просмотрим еще раз весь набор i.
- Если точки (x_1, y_2) и (x_2, y_1) существуют в наборе i, поддерживаем максимальное значение и координаты точек.

Вариант эффективного по времени решения.

Создадим для точек каждой стороны по дополнительному двумерному массиву M_1 и M_2 размером (2·MAX_COORD+1)×(2·MAX_COORD+1), где MAX_COORD=1000. Если точка (x, y) находится на первой стороне, то значение [x+MAX_COORD][y+MAX_COORD] в массиве M_1 будет иметь значение 1, остальные элементы равны 0.
Также заполняется массив $M2$, если точка (x, y) находится на второй стороне, то значение [x + MAX_COORD][y + MAX_COORD] в массиве M_2 будет иметь значение 1, остальные элементы равны 0.

Далее необходимо решить задачу нахождения прямоугольника наибольшей площади для точек с каждой стороны по отдельности:

- Попарно переберем все точки со стороны i (i=1,2).
- Для пары точек (x_1, y_1) и (x_2, y_2) проверим можно ли образовать прямоугольник, в котором эти точки будут диагонально противоположны. Две точки образуют такой прямоугольник, если на стороне i существуют точки (x_1, y_2) и (x_2, y_1). Проверить это можно, обратившись к элементам [x_1 + MAX_COORD][y_2+MAX_COORD] и [x_2+ MAX_COORD][y_1+MAX_COORD] в массиве M_i.
- Если прямоугольник образуется, считаем его площадь как $|x_1 - x_2| \cdot |y_1 - y_2|$.
- Из всех полученных площадей поддерживаем максимальное значение и координаты четырех точек, образующих прямоугольник с максимальной площадью.

Задача 5. Сбор шишек

Обозначим $values[i]$ как массив, в котором хранится количество шишек у каждого кедра.
Для решения задачи будем поддерживать динамику следующего вида:

- заполним все состояния динамики большим числом INF;

- пусть $dp[i][j]$ — минимальная усталость медведя, которую он может достичь, если он прошел кедр с номером i, и собрал количество шишек равное числу j;
- база динамики – $\begin{cases} dp[0][0] = 0 \\ dp[0][values[0]] = 0 \end{cases}$;
- переходы динамики:
  - считаем $dp[i][j]$;
  - рассмотрим все состояния с предыдущего шага. Если $dp[i-1][j] \neq INF$, то можно обновить $dp[i][j] = dp[i-1][j] + j$. Это означает, что медведь набрал j шишек возле предыдущих кедров и потратил $+\ j$ усталости, чтобы совершить переход до кедра с номером i;
  - обновить также нужно значения динамики следующим образом: $dp[i][j] = \min\ (dp[i][j], dp[i-1][j - values[i]] + (j - values[i])$. Тем самым мы пытаемся улучшить результат, пробуя добавлять шишки от текущего кедра;
- ответом является максимум среди всех j, для которых значение динамики не превосходит параметр K;
- итоговая асимптотика алгоритма: $O(N \cdot MAXS)$, где $MAXS$ — максимальное количество шишек (10^4).

Частичное решение, неэффективные по времени, — полный перебор кедров, с которых собираются шишки.

Частичное некорректное решение — брать шишки со всех кедров по порядку, пока не превысим параметр K.