Задача 5. Треугольники

Подгруппа 1

В данной подгруппе возможны только треугольники со сторонами 1, всего палок со стороной 1 будет $k + 1$, так как у Васи тоже есть палка со стороной 1.

Подгруппа 2

При $k = 1$ можно составить один равнобедренный треугольник только, если $d \leqslant n$, иначе все палки будут разного размера и нельзя будет вообще составить треугольники.

При $k = 2$ всегда можно составить только 1 равнобедренный треугольник.

При $k = 3$ всегда можно составить 1 равнобедренный треугольник. И еще 1 дополнительный, если $d \leqslant n$.

Подгруппа 3

Все палки Васи могут быть только основаниями, пытаемся достроить их до треугольников, используя палки Пети. Из оставшихся палок у Пети, если они остались, можно будет построить еще равносторонние треугольники.

Подгруппа 4, 5, 6

Подгруппы для различных переборных решений.

Полное решение

Палки Васи могут быть только основанием равнобедренного треугольника, так как встречаются по одному разу, кроме, возможно, палки d, если она у него есть. Если $d \leqslant n$, давайте передадим палку длины d Пете, тогда у Пети станет $k = k + 1$палка, а у Васи $n = n - 1$.

Теперь нужно узнать, а сколько из палок Васи могут быть основаниями, если боковыми сторонами будут палки длины d.

Если $2 \cdot d - 1 \geqslant n$, то все палки Васи могут быть основаниями, иначе только первые $2 \cdot d - 2$ могут быть основаниями, так как остальные палки будут длиннее суммы двух боковых сторон.

Таким образом мы посчитали, сколько палок у Васи могут быть основаниями, остается посчитать, хватит ли палок у Пети, чтобы достроить основания Васи до равнобедренных треугольников. Т.е. строим $min(x, \frac{k}{2})$ треугольников, где x — сколько палок у Васи могут быть основаниями.

И последний шаг — построить равносторонние треугольники из оставшихся палок Пети.

Задача 6. Камень, ножницы, бумага

Подзадача 1

Строка s состоит из символов r. Оптимальным ответом будет строка, в которой, например, чередуются символы p и r (именно в таком порядке).

Подзадача 2

Во входной строке s все соседние символы различны. Это значит, что в выходной строке при построении оптимального ответа соседние символы также будут различными, каждый ход Васи будет выигрышным.

Подзадача 3

Маленькие ограничения на n позволяют перебрать все возможные строки в качестве ответа. На каждую позицию можно поставить 3 различных символа. А значит всего возможных строк 3^n. Среди всех строк нужно выбрать ту, количество побед в которой максимально.

Подзадача 4

В четвертой подзадаче строка состоит только из символов r и s. В начале каждой подстроки из подряд идущих букв r поставим букву p, а затем, пока такая подстрока не завершилась, будем чередовать буквы s и p. А каждую подстроку из подряд идущих букв s будем начинать с буквы r и продолжать чередованием букв p и r. Можно доказать, что построенный таким образом ответ будет оптимальным.

Подзадача 5

Идея решения пятой подзадачи основывается на идее полного решения, описанного в шестой подзадаче, но допускает любые другие реализации, имеющие большее время работы.

Подзадача 6

Для полного решения задачи можно использовать жадные алгоритмы или идею динамического программирования. Здесь будет разобрана идея жадного подхода.

Для начала, не обращая внимания на повторы, проставим на все позиции в строке символы выигрышных жестов. Затем выполним проход по строке слева направо. В том случае, если текущий символ совпадает с предыдущим, будем заменять текущий символ так, чтобы он не был равен ни предыдущему, ни следующему. Поскольку ничья приравнивается к поражению, результат будет оптимальным.

Задача 7. Ботать!

Заметим, что если нет ромашек с нечетным количеством лепестков, то у нас всегда будет 0 вариантов. Иначе мы можем комбинировать и составлять последовательности, где должна быть хотя бы одна ромашка с нечетным количеством лепестков.

Подгруппа 1, 2

Можно руками попробовать рассмотреть всевозможные последовательности и вывести ответ.

Подгруппа 3, 4, 5

В этой подгруппе все последовательности будут подходящими, остается правильно посчитать их количество.

Сведем задачу к следующей — для каждого k от 1 до n нам нужно выбрать какие-то k ромашек из n в различном порядке. Это базовая комбинаторная задача — найти количество размещений A_n^k.

Подгруппа 6, 7

Данные подгруппы можно решить различными способами, например, при помощи динамического программирования.

Подгруппа 8 и полное решение

Заметим следующее — чтобы посчитать количество подходящих последовательностей, можно из общего количества вычесть количество неподходящих последовательностей. Неподходящие последовательности состоят только из ромашек с четным количеством лепестков.

Пусть x — количество ромашек с четными лепестками. Тогда всего вариантов — сумма всех A_n^k для k от 1 до n. А неподходящих — сумма всех A_x^k для k от 1 до x.

Для полного решения удобно предпосчитать факториалы и обратные факториалы для всех k от 1 до n, а затем за $O(1)$ вычислять каждое необходимое A_n^k, а не вычислять каждый раз линейно.

Задача 8. Атаки феникса

Во многих подзадачах удобно считать количество атак следующим образом: посчитать суммарный нанесенный усиленными атаками урон D, после чего потребуется еще $H - D$ атак с уроном 1, где H — сумма значений h_i. Кроме того верно утверждение: усиленной атакой выгодно атаковать существо, у которого на данный момент наибольшее количество здоровья. При этом можно считать, что сначала проходят все усиленные атаки, а только затем атаки с уроном 1.

Подзадача 1

Поскольку у Васи маны нет, то все атаки будут наносить 1 единицу урона, поэтому нужно лишь найти сумму значений h_i.

Подзадача 2

В этой подзадаче при усиленной атаке вражеский юнит всегда погибает, поэтому усиленными атаками нужно уничтожать юнитов с наибольшим количеством здоровья. Всех оставшихся существ (если $n < m_i$) придется уничтожать атаками силы 1, для этого опять нужно найти их суммарное количество здоровья. Для того, чтобы на каждый запрос можно было отвечать достаточно быстро, не перебирая всех существ, можно воспользоваться префикс-суммами: сохранить суммарное значение здоровья $1, 2, \ldots, n$ юнитов с наименьшим значением h_i.

Подзадача 3

Поскольку все h_i равны, то усиленная атака может нанести d единиц урона, если осталось хотя бы одно существо с достаточным количеством здоровья, либо $d_1 = h_i mod d$ (остаток от деления). Легко посчитать, какое количество атак можно провести с уроном d и с уроном d_1. Тогда для фиксированного значения m_i сначала надо проводить усиленные атаки с уроном d, затем — с уроном d_1. Таким образом, посчитать суммарный нанесенный усиленными атаками урон можно за $O(1)$ операций.

Подзадача 4

В этой подзадаче усиленные атаки могут наносить урон 2 или урон 1. По аналогии с предыдущей подзадачей имеет смысл посчитать, какое наибольшее количество атак с уроном 2 может нанести феникс. Далее аналогично за $O(1)$ можно найти суммарный нанесенный усиленными атаками урон.

Подзадача 5

Здесь для каждого запроса можно независимо промоделировать поведения феникса: каждый раз атаковать врага с наибольшим здоровьем, пока есть мана на усиленные атаки. При этом можно находить это существо простым поиском за $O(n)$, всего же таких операций для одного запроса будет не более $O(n \cdot h_i)$, итого общая сложность алгоритма будет $O(n^2 \cdot h_i \cdot q)$.

Подзадачи 6 и 7

Для каждого существа надо посчитать значение k_i: сколько атак можно ему нанести с полным уроном d. Если все эти атаки пройдут, то у существа останется $r_i = h_i - d \cdot k_i$ единиц здоровья. Понятно, что после проведения K атак с полным уроном (где k — сумма значений k_i) следующие атаки нужно проводить по существам с наибольшим значением r_i. Тогда ещё до обработки запросов нужно упорядочить существ по значению r_i. Для подзадачи 6 достаточно для каждого запроса посчитать значение t — количество оставшихся усиленных атак, после чего найти сумму наибольших t значений r_i за $O(n)$. Общая сложность алгоритма будет $O(n \cdot ln(n) + q \cdot n)$. Для подзадачи 7 нужно заменить такой поиск на префикс-суммы по аналогии с подзадачей 2. Тогда вычислительная сложность будет $O(n \cdot ln(n) + q)$.