Задача 1. Вкусняшки (7-8 кл.)

В данной задаче нужно было отсортировать все «вкусняшки», найти текущую позицию Бобика относительно получившегося массива и аккуратно просимулировать процесс поедания вкусняшек.

Задача 2. Преобразование карты (7-8 кл.)

Необходимо создать таблицу из (180 / h) строк и (360 / w) столбцов, заполнить ее нулями.

Для всех городов по порядку вычислить их новые координаты и поместить их в таблицу на нужное место. Если в клетке, куда мы хотим записать очередной город стоит не ноль, значит, ответ на задачу отрицательный, и мы можем вывести в качестве ответа номера городов — тот, что записан в таблице и тот, который мы на это место хотим записать.

В процессе записи города в таблицу нужно отмечать номер строки и номер столбца, как непустые.

Далее, можно пересчитать координаты городов с учетом пустых столбцов и строк.

Задача 3. Головоломка (7-11 кл.)

Назовем состояние головоломки, когда стрелки указывают на одно деление, хорошим.

Решение головоломки можно разбить на два этапа:

1. перемещение стрелок до некоторого хорошего состояния,

2. прохождение всех хороших состояний из некоторого хорошего состояния.

Можно заметить, что на втором этапе не важно, на каком конкретно делении стоят стрелки к моменту началу этого этапа. Поэтому количество действий для каждого этапа можно минимизировать отдельно.

Посчитаем минимальное количество действий для первого этапа. Пусть расстояние между стрелками — это то, на сколько делений по часовой стрелке надо повернуть первую стрелку, чтобы стрелки указывали на одно деление. Тогда после действия, в котором движется одна стрелка, $\rho_2 = (\rho_1 \pm 1) \mod N$, где ρ_2 — расстояние между стрелками после действия и ρ_1 — расстояние до действия. После движения обеих стрелок в одном направлении это расстояние не меняется, а если они передвигаются в разном направлении, то $\rho_2 = (\rho_1 \pm 2) \mod N$.

В хорошем состоянии это расстояние равно 0. А значит, для простой головоломки количество действий для самого быстрого перехода в хорошее состояние равно $\min(|f-s|, N-|f-s|)$, а для сложной версии равно $min(\left\lfloor\frac{|f-s|}{2}\right\rfloor + |f-s| \mod 2, \left\lfloor\frac{N-|f-s|}{2}\right\rfloor + (N-|f-s|) \mod 2)$.

Теперь посчитаем минимальное количество действий для прохода по всем хорошим состояниям, если в начале мы находимся в хорошем состоянии.

Для простой версии это количество равно $N-1$, так как при сдвиге обеих стрелок в одном направлении из хорошего состояния переходим в хорошее. То есть, в случае простой версии надо просто $N-1$ раз передвинуть обе стрелки по часовой.

Для усложненной версии при $N \geqslant 5$ количество действий равно $3 \cdot (N-1)$, так как при таком N нельзя меньше, чем за три хода, перейти из хорошего состояния в другое хорошее состояние. А за три хода можно перейти в следующее по часовой стрелке хорошее состояние. Для этого надо:

1. сдвинуть первую стрелку против часовой, а вторую по часовой,

2. сдвинуть первую стрелку по часовой,

3. еще раз сдвинуть первую по часовой.

Количество действий для $N \leqslant 4$ можно получить перебором.

Итоговым ответом будет сумма количества действий для каждого этапа. Итоговая асимптотика решения — $O(1)$.

Задача 4. Крутилки и вертелки (7-11 кл.)

В данной задаче нужно было построить граф по шестеренкам: каждая шестеренка — это вершина графа. Если две шестеренки касаются друг друга, то между соответствующими вершинами добавляем ребро.

Когда граф построен, его надо обойти используя *DFS* или *BFS*, крася вершины поочередно в белый и черный цвета. Если получилось так сделать, то значит, черная вершина соединена только с белыми, и наоборот. Следовательно, граф не имеет циклов нечетной длины, и ответ на задачу `YES`. В противном случае — ответ `NO`.

Сложность алгоритма: $O(N^2)$.

Задача 5. VR игра (7-11 кл.)

Рассмотрим уравнение параболы $y = -a \cdot x \cdot (x - x_0)$, где x_0 — конечная точка траектории мяча. Неизвестным является коэффициент a.

Для решения второй группы тестов достаточно перебрать коэффициент a с некоторым шагом, например, 10^{-3} и проверять, какие кольца будет пересекать парабола с заданным a. Границами перебора необходимо брать минимальный и максимальный коэффициент a, при котором парабола будет пересекать хотя бы одно кольцо. Для того, чтобы их найти, достаточно перебрать все кольца и посчитать a на концах отрезка.

Заметим, что не нужно перебирать все возможные a, которые проходят через кольца, а достаточно проверить только такие a, которые получаются, когда парабола проходит через нижнюю точку отрезка, являющегося проекцией кольца. Таких точек ровно N, проверим для каждой, какое количество колец будет пересекать парабола, и получим решение с итоговой асимптотикой $O(N^2)$. Такое решение было достаточно, для того чтобы прошли тесты из третьей группы.

Можно увидеть, что для каждого кольца существует отрезок $[a_{i_{min}}, a_{i_{max}}]$, на котором парабола с любым коэффициентом из этого отрезка его пересекает. Построим массив, содержащий все концы таких отрезков и отсортируем его. Будем совершать обход этого массива и, если встречаем левый конец отрезка, то добавим единицу к ответу, а если правый — отнимем ее. Таким образом, необходимо найти точку в массиве, в которой ответ максимальный, и восстановить номера колец, которые будут пересечены такой параболой. Итоговая асимптотика этого решения — $O(N \cdot logN)$.

Необходимо уточнить, что ограничения на координаты позволяют находить, хранить и сравнивать коэффициенты как несократимые дроби, таким образом, решая задачу в целых числах. В задаче это необходимо для получения точного ответа и ускорения решения.

Задача 6. Белуга и Скиттл (7-11 кл.)

Для решения первых трёх подгрупп достаточно явно сопоставить шаблон амогуса с изображением во всех позициях.

Для решения последней подгруппы нужно посчитать хеш от всех прямоугольников размера $A \times B$ в изображении и, если значение хеша совпадает со значением хеша прямоугольника, увеличить ответ. В качестве хеш-функции можно использовать полиномиальный хеш по простому модулю.

Задача 7. Арсений и удаление файлов (9-11 кл.)

Для решения 2-й подгруппы можно было для каждого запроса на изменение проходить по всем дочерним и родительским объектам обновляемой вершины, обновляя их состояние. Сложность такого решения — $O(Q \cdot N)$.

Для решения 3-й подгруппы можно было хранить только состояния папок. А также для каждой папки хранить список «выколотых» файлов, у которых состояние отличается от родителя (например, в `std::set`). Если «выколотых» файлов больше 0, состояние папки равно 1. Если все дочерние файлы становятся «выколотыми», их можно обнулить и поменять состояние родительской папки на противоположное. Сложность такого решения — $O(Q \cdot \log N)$.

Для полного решения заметим, что если занумеровать файлы в порядке их обхода в глубину, то каждой папке будет соответствовать непрерывный отрезок файлов, которые являются дочерними для этой папки. Также заметим, что состояние любой папки определяется тем, сколько дочерних файлов (на любом уровне) под ней выделено. Если выделены все файлы, то состояние равно 2, если ни один файл не выделен, состояние равно 0, иначе — состояние равно 1. Заведем дерево отрезков на файлах, где у каждого файла будет состояние 0 или 1, в зависимости от того, выделен он или нет. Будем выполнять присвоение на отрезке, соответствующем папке или файлу, при выделении или снятии выделения (файлам соответствуют отрезки длины 1). Для определения состояния будем считать сумму на отрезке. Сложность такого решения — $O(Q \cdot \log N)$.

Задача 8. Две фотографии (9-11 кл.)

Для решения первой группы можно явно перебирать позиции начала второй ломаной и проверить совпадение с первой.

Для решения второй подгруппы заметим, что части ломаных будут совпадать тогда и только тогда, когда подстроки разностей высот равны. Для решения достаточно построить две строки — разностей высот соседних вершин в ломаной и проверить, является ли вторая строка подстрокой первой.

Для решения остальных подгрупп заметим, что одну ломаную можно будет сопоставить с другой, если отношения длин проекций на оси X и Y совпадают у обеих ломаных. Можно дополнить решение второй подгруппы, строя вместо строк разниц по оси Y, отношением координат текущего отрезка к координатам предыдущего отрезка.

Задача 9. Спецагент Петя (9-11 кл.)

Для решения первой подгруппы достаточно рассмотреть пути, в которых Петя зацепляется за ровно один столб. Точками касания касательных из точки (x, y) к окружности с центром в $(0, 0)$ и радиуса R будут точки $(r \cdot \cos(\alpha + \beta), r \cdot \sin(\alpha + \beta))$ и $(r \cdot \cos(\alpha - \beta), r \cdot \sin(\alpha - \beta))$, где $\alpha = atan2(y, x)$, а $\beta = acos\left(\frac{r}{\sqrt{x^2 + y^2}}\right)$. Остаётся только аккуратно рассмотреть правильные пути между начальной позицией, конечной позицией с учетом движения по и против часовой стрелки.

Для следующих подгрупп заметим, что между любыми двумя окружностями четыре касательных — две внешние и две внутренние. Внешние касательные можно построить путём параллельного переноса отрезка, соединяющего центры окружностей, на R. Поскольку все радиусы равны, то внутренние касательные соответствуют касательным, проведённым к окружностям из середины отрезка, соединяющего их центры.

Для второй и третьей подгрупп достаточно выкинуть из рассмотрения касательные, которые пересекают препятствия. После чего построим полный взвешенный граф на точках касания и найдём кратчайший путь от начальной позиции до конечной.

Для решения четвертой подгруппы нужно заметить, что при пересечении окружности препятствием, часть точек касания перестаёт быть достижимой по дуге из остальных. Для того, чтобы удалить лишние рёбра из графа, отсортируем точки касания по полярному углу относительно центра окружности. Тогда точки пересечения разбивают точки касания на «группы» достижимых. Рёбра между точками касания существуют тогда и только тогда, когда точки находятся в одной группе.

Для решения последней подгруппы заметим, что хранить ребра между каждой парой точек касания в группе не нужно, достаточно простроить ребра между соседними точками в группе (соседними с точки зрения полярного угла).