

## Разбор задачи «Диманово»

Решение на 60 баллов: для каждой точки переберем все отрезки, найдем ответ (точка должна лежать на отрезке и длина максимальна). Получим решение за  $O(nm)$ .

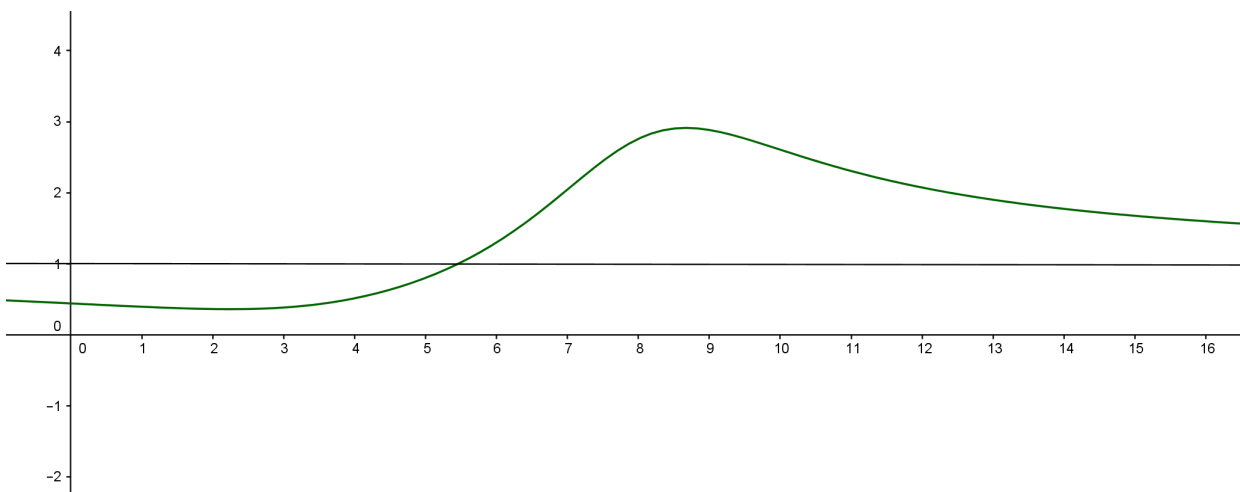
Так как координаты точек большие, то воспользуемся методом сжатия координат: отсортируем все координаты и занумеруем их в таком порядке, затем будем использовать их номера вместо самой координаты. Легко понять, что номер координаты не превысит  $10^5$ . Затем, построим дерево отрезков на них. Отсортируем отрезки по длине (по неубыванию), если длины равны, то раньше должен идти отрезок с большим номером. Пройдемся в этом порядке по отрезкам и будем делать присвоение на отрезке индексом отрезка. Это можно делать с помощью запаздывающих обновлений. В самом конце для каждой камеры посмотрим в дереве отрезков, какое значение ей присвоено последнее, и выведем его. Описание алгоритмов построения и использования структуры «дерево отрезков» можно найти в интернете на популярных ресурсах, посвященных олимпиадным задачам.

## Разбор задачи «Стрелки»

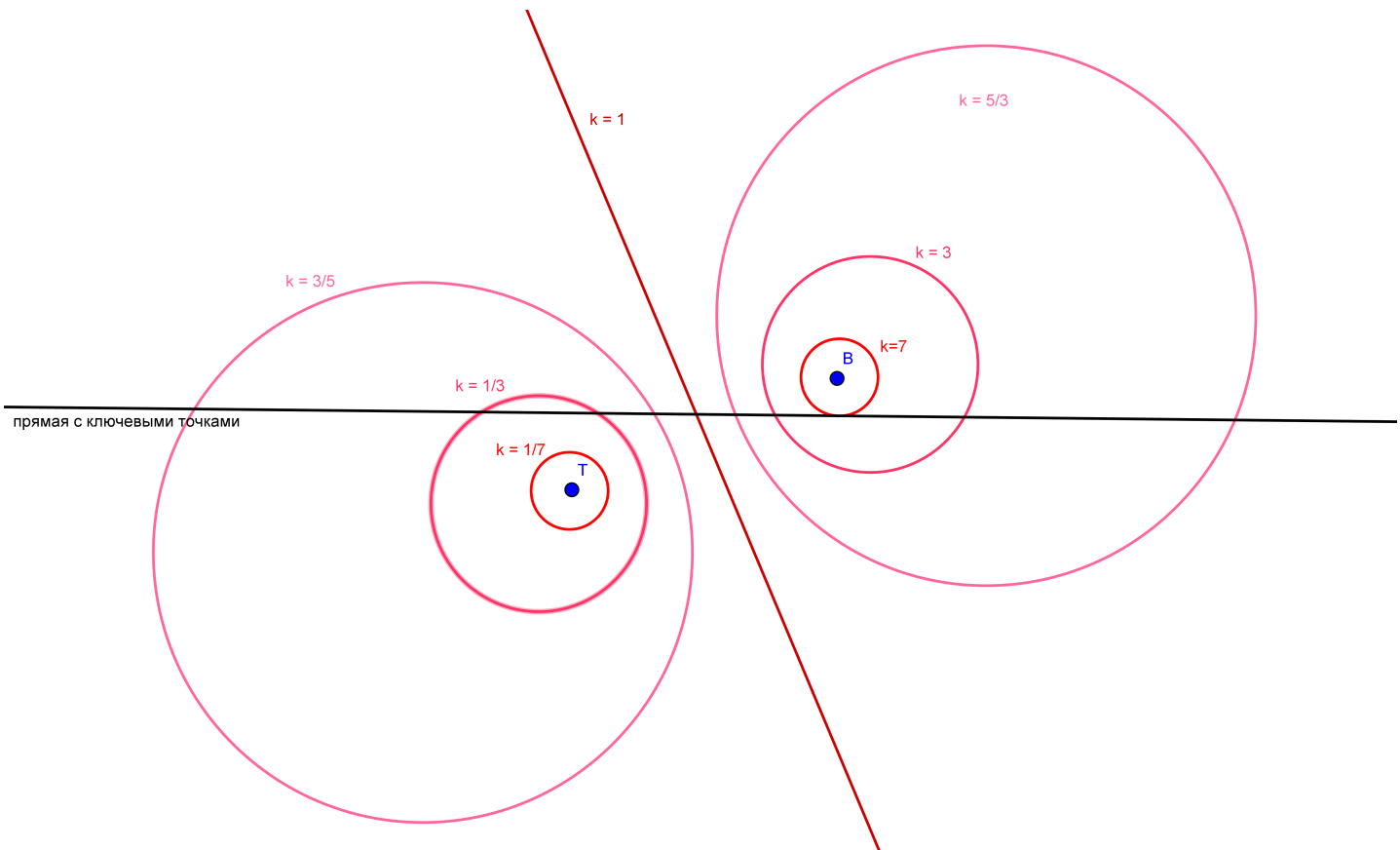
Пусть Тарас находится в точке  $T$ . Для каждого разбойника  $B$  переберем все ключевые точки  $K$ . Можно определить минимальную скорость, которую должна иметь стрела Тараса, чтобы прилететь в точку  $K$  не позже стрелы разбойника. Эта скорость —  $\frac{TK}{BK}v$ , где  $v$  — скорость разбойника. Теперь, максимум по всем таким скоростям и будет ответом. Такое решение работает за  $O(nq)$  времени, что, конечно, чрезвычайно долго.

Рассмотрим более оптимальное решение. Заметим, что дорога узка — всего 101 различная координата по оси  $y$ . Разобьем все ключевые точки на классы по равенству  $y$ -координаты. Будем обозначать максимальную  $y$ -координату по всем ключевым точкам  $y_{max}$ . Разбиение и сортировка обойдутся нам в  $O(q \log q)$  времени.

Теперь можно перебрать для каждого разбойника  $B$  целый класс ключевых точек. Осталось понять, как в классе ключевых точек найти такую точку  $K$ , что отношение  $\frac{TK}{BK}$  было бы наибольшим. Все точки внутри одного класса лежат на горизонтальной прямой. Можно исследовать вид функции  $\frac{TK}{BK}$ , зависящей от  $x$ -координаты точки  $K$ . Несложно понять, что при удалении точки  $K$  на бесконечность отношение будет стремиться к 1. Само же значение 1 может быть достижимо только на срединном перпендикуляре отрезка  $TB$ . Слева и справа от значения 1 функция будет иметь точку максимума и точку минимума.



Как показать, что функция будет иметь точки локального максимума и минимума? Можно выписать отношение  $\frac{TK}{BK}$  в координатном виде и приравнять производную выражения к 0. Получится квадратное уравнение, имеющее два корня — это и есть точки максимума и минимума. Есть и другой путь. Заметим, что ГМТ точек таких, что  $\frac{TK}{BK} = k$  есть окружность Аполлония, либо прямая при  $k = 1$ . Горизонтальная прямая пересекает эти окружности в двух точках, но только одну из окружностей касается. В это окружности будет точка минимума(максимума).



Как же теперь решать задачу? Бинарным поиском разделим точки на прямой на два множества. Для первого множества (для определенности находится слева) выполнено  $\frac{TK}{BK} \leq 1$ , для второго (справа)  $\frac{TK}{BK} > 1$ . Если второе множество пусто, то Тарасу не нужна скорость большая скорости рассматриваемого разбойника. Вспомним, что максимум  $\frac{TK}{BK}$  в левой части находится на концах отрезка. Тогда просто проверим концы отрезка и обновим ответ. Теперь предположим, что второе множество не пусто, тогда первое множество рассматривать не надо совсем. Ответ в правом множестве есть точка максимума. Можно найти ее тернарным поиском по ответу, а можно и бинарным по дискретной производной. Тогда итоговая оценка времени работы  $O((ny_{max} + q) \log q)$ .

## Разбор задачи «Ярмарка»

По условию нас интересуют только товары с ярлыком «Суперцена», а значит, если мы зафиксируем позиции минимумов, то остальные элементы можем расставить произвольно.

Пусть  $a = a_1, a_2, \dots, a_n$  - ответ на задачу. Тогда какой-то из элементов  $a_1, a_2, \dots, a_k$  будет равен минимуму. Давайте покажем, что оптимально поставить минимум в качестве элемента  $a_k$ . Элемент  $a_k$  входит в отрезки  $[a_1, \dots, a_k]$ ,  $[a_2, \dots, a_{k+1}]$ , ...,  $[a_k, \dots, a_{2k-1}]$ . Если мы вместо  $a_k$  выберем некоторый другой элемент из  $a_1, \dots, a_{k-1}$ , то количество отрезков, которым принадлежит этот минимум, может только уменьшиться. Рассуждая аналогичным образом получаем, что оптимальным будет поставить минимумы на позиции  $k, 2k, 3k$  и т.д. Если в какой-то момент нам нужно поставить элемент, а минимумы закончились, значит ответ - Impossible.

## Разбор задачи «Vomberman на дереве»

Сначала вычислим таблицу  $d[i][j]$  кратчайших расстояний от каждой вершины до каждой (например, можно сделать это за  $O(N^2)$ , запустив поиск в ширину из каждой вершины). С помощью этой таблицы найдём, какое нужно время, чтобы дойти до  $a_1$  (первого бункера, который надо взорвать). Теперь мы закладываем бомбу в этот бункер и двигаемся в следующий  $a_2$  за минимальное возможное время так, чтобы нас не задело взрывом. Как же нам его вычислить? Для этого переберём номер вершины  $k$ , в которой мы находимся в момент взрыва. С одной стороны,  $d[a_1][k]$  должно

быть больше  $p$  (чтобы мы не пострадали). А с другой стороны  $d[a_1][k] \leq t$  (иначе мы не успеем добежать в эту вершину). Понятно, что из всех вершин, подходящих под данное условие, мы должны выбрать ту, у которой  $d[k][a_2]$  минимальное. Суммарное время на подрыв вершины  $a_1$  и путь до следующей будет занимать  $t + d[k][a_2]$ . Ясно, что если таких вершин  $k$  нет, то наша миссия обречена на провал (нам просто негде скрыться от взрыва).

Теперь проделаем эти вычисления для всех остальных пар бункеров ( $a_2$  и  $a_3, \dots, a_{m-1}$  и  $a_m$ ).

С последним бункером надо разобраться отдельно: также переберём вершину  $k$ , в которой мы будем прятаться в момент взрыва. Если она существует, то время, которые мы истратим на последний бункер, равно  $t$ . А если её не существует – то в любом случае мы будем задеты взрывом. Итоговый ответ будет равен сумме всех вычисляемых нами времен или равен  $-1$  в случае, если на каком-то этапе провал неизбежен.

## Разбор задачи «Спецзадание»

Пусть у нас есть только один охранник. В таком случае, это стандартная задача на поиск в ширину: мы добавляем в очередь позицию этого охранника, помечаем расстояние до этой позиции равным нулю, далее, добавляем в очередь все смежные с этим охранником позиции и определяем расстояние равным единицы, и так далее. В конце получаем расстояние от этого охранника до всех позиций.

Для того, чтобы набрать 60 баллов, достаточно было производить поиск в ширину последовательно для каждого охранника. Время работы такого решения:  $O(n^2m^2)$ .

Чтобы улучшить время работы, попробуем запустить поиск в ширину для всех охранников одновременно. Мы можем сделать точно такой же поиск в ширину, но теперь изначально в очередь нужно добавить позиции всех охранников и проинициализировать расстояния до них нулями. Далее, выполняем обычный поиск в ширину, таким образом, сначала мы добавим в очередь все позиции, смежные с первым охранником, потом со вторым и т. д. Итоговая асимптотика:  $O(nm)$ .