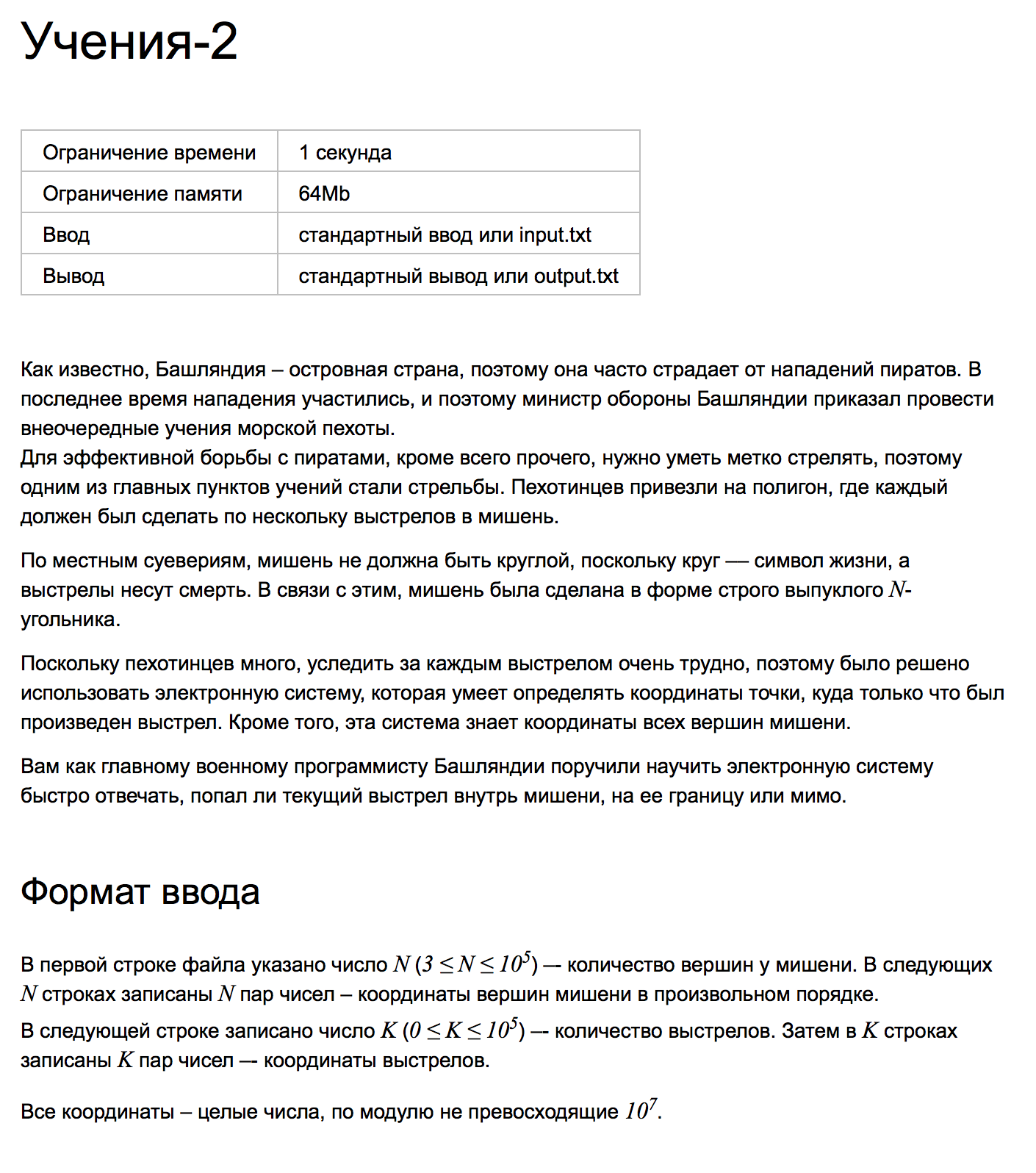
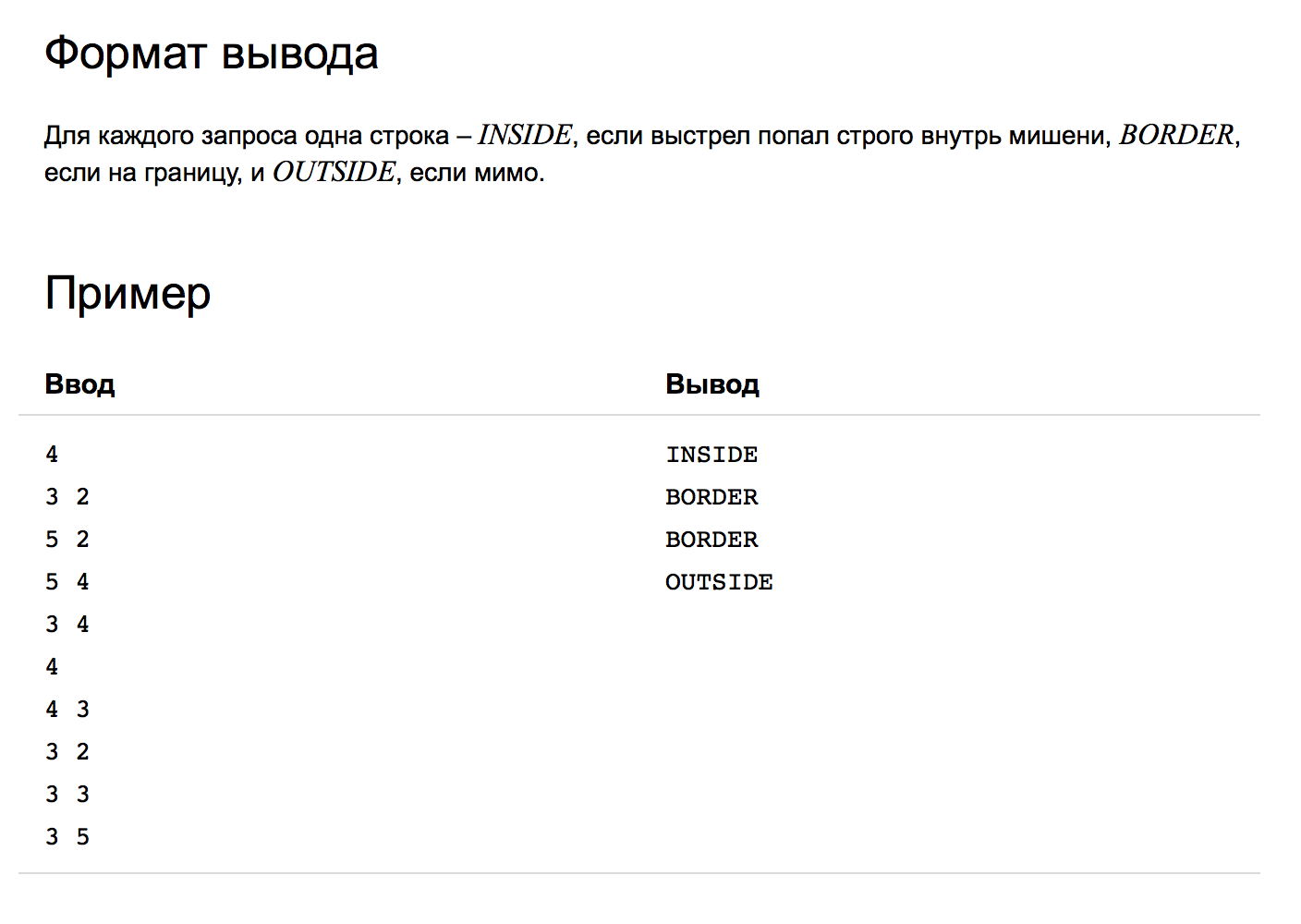
Задача «Учения-2»

Решение Евченко Богдана ПИ-15-1

<https://github.com/olderor/advanced7>





Решение.

1. Отсортируем точки, которые задают мишень, в порядке обхода против часовой стрелки, начиная с случайно выбранной.
2. Зафиксируем первую точку.
3. Проверим, лежит ли точка на двух отрезках, каждый из которых задается выбранной точкой и ее соседом слева или справа соответственно. Если лежит – ответ BORDER, точка лежит на границе мишени.
4. Проведем два луча через эти два отрезка, начиная с выбранной точки. Плоскость разделится на две части – часть с мишенью и часть без.
5. Проверим, в какой из частей лежит точка. Это можно сделать вычислив направление поворота для точки, относительно вектора. Если точка лежит в части без мишени – значит выстрел мимо и ответ OUTSIDE, иначе продолжаем вычисления.
6. Применим принцип бинарного поиска. Для этого возьмем часть с мишенью и разделим ее пополам, относительно вершины, которая находится посередине в списке точек (проведем луч от изначальной точки в сторону этой вершины). Аналогичным образом (как в 4 и 5) проверим, находится ли точка внутри первой части или второй.
7. Будем разбивать эти части пополам, пока не найдем сегмент, который состоит из двух лучей, проведенных из зафиксированной точки (первый в сторону какой-либо точки и второй в сторону ее соседа, назовем их p1 и p2).
8. Проверим, лежит ли точка на отрезке p1-p2. Если лежит - ответ BORDER.
9. Проверим, пересекаются ли отрезок p1-p2 и отрезок начинающийся с зафиксированной точки, заканчивающийся точкой-выстрелом. Если пересекаются, ответ OUTSIDE, иначе - INSIDE

Доказательство

Поскольку все точки формируют выпуклую оболочку, то, выбрав изначальную точку, их можно отсортировать, сравнивая значение поворота для одной из двух сравниваемых точек относительно вектора, начинающегося в изначальной точке, заканчивающимся в другой из сравниваемых. (Примечательно, что поменяв местами сравниваемые точки, можно изменить порядок обхода точек оболочки на почасовую и наоборот).

Пункт 3 решения очевиден, потому что точка не может одновременно лежать внутри или на границе мишени и части, где нет мишени. Заметим, что пункт для границы мы проверили отдельно, и точка не может лежать на других ребрах, иначе бы она находилась внутри или соприкасалась с частью мишени, что противоречит выясненному.

Применяя идею бинпоиска одну из частей фигуры можно спокойно не рассматривать, поскольку это очевидно, что если точку видно из одного сегмента, то она никак не может попасть в часть второго. Единственный вариант, когда точка находится на середине (общей части) двух частей, тогда из двух частей можно взять любую, поскольку точка будет лежать на луче, начинающемся в зафиксированной точке и проходящий через текущую точку, и ответ не будет зависеть от того, с какой стороны к ней подходить – слева или справа, в конце концов все равно будет рассматриваться именно этот луч.

Рассмотрим случаи, когда мы нашли интересующий сектор.

Если точка лежит на отрезке от зафиксированной точки до p1, или на отрезке от зафиксированной точки до p2, то она не может находится на ребре, поскольку единственные соседи зафиксированной точки были проверены в самом начале алгоритма, а значит если точка лежит на каком-то из этих отрезков – значит она лежит внутри мишени, что в дальнейшем не повлияет на результат. Единственный оставшийся вариант – когда точка лежит на ребре p1-p2 был также отдельно проверен, значит все возможные случаи, когда точка лежит на границе, были рассмотрены.

Поскольку было выяснено, что точку видно из текущего сектора, она не находится ни справа, ни слева, значит возможны всего два случая – либо точка лежит внутри фигуры (в данном случае треугольника с вершинами в зафиксированной точке, p1 и p2), либо снаружи. Это можно легко и однозначно проверить, проверив пересекаются ли два отрезка: отрезок p1-p2 и отрезок начинающийся с зафиксированной точки, заканчивающийся точкой-выстрелом.

Итого, для каждого возможного случая было определено расположение точки, что дает однозначный ответ на запрос.

Итоги

Оценка памяти. Для решения задачи потребовалось хранить точки, формирующие мишень , хранить все точки, которые представляют из себя запросы-выстрелы и ответ для каждого выстрела

Оценка времени работы. Для решения задачи необходимо считать точки за. Отсортировать точки за найти ответ бинпоиском за для каждого запроса, что в целом составляет , сформировать результат за

Итог:

* затраты по памяти.
* затраты по времени.