

Задача А. Картинная галерея

Имя входного файла: gallery.in
Имя выходного файла: gallery.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В картинной галерее, имеющей форму N -угольника, расположено M люстр, которые мы будем считать точечными источниками света. Точка стены галереи называется освещенной, если из нее видна хотя бы одна из люстр. Неосвещенным участком будем называть максимальное связное множество точек стены галереи, ни одна из которых не освещена (участок может содержать углы галереи). Напишите программу, определяющую все неосвещенные участки.

Идея решения: Рассмотрим решение задачи, если источник света один. Разобьем все стороны N -угольника на отрезки так, чтобы каждый отрезок был либо целиком освещен, либо целиком неосвещен. Для этого проведем из источника света лучи через те вершины N -угольника, угол при которых больше 180° , и найдем ближайшие точки пересечения этих лучей со сторонами N -угольника (касания не рассматриваются). Эти точки и дадут нам требуемое разбиение. Теперь для каждого отрезка надо проверить, освещен он или нет. Проведем луч из источника света через середину проверяемого отрезка. Найдем точки пересечения этого луча со сторонами N -угольника и среди них отыщем наименее удаленную от источника света. Если это середина нашего отрезка, то отрезок освещен, если нет, то та сторона, которой принадлежит эта точка, закрывает проверяемый отрезок от источника света. Проведем эту процедуру для остальных источников света, разбивая имеющиеся неосвещенные отрезки на более мелкие (для освещенных отрезков это бессмысленно). При выводе соседние неосвещенные участки надо объединить.

Формат входного файла

Первая строка входного файла содержит два целых числа N и M ($1 \leq N, M \leq 30$). В каждой из следующих N строк записаны координаты очередного угла галереи. Углы перечислены в порядке обхода стены по часовой стрелке. Далее идут M строк, каждая из которых содержит координаты очередной из люстр. Все координаты являются вещественными числами и разделяются пробелом.

Формат выходного файла

В первую строку выходного файла выведите количество неосвещенных участков S . Каждая из следующих S строк должна содержать описание очередного из участков в виде тройки чисел, разделенных пробелом. Первые два числа определяют координаты начальной точки участка, третье - его длину. (Участок должен продолжаться на указанную длину в направлении обхода стены по часовой стрелке. Никакие два участка не должны иметь общих точек.) Числа, определяющие участок, должны быть выведены не менее чем с 3 верными значащими цифрами.

Примеры

gallery.in	gallery.out
5 1	1
0 0	1 5 5.82843
0 5	
4 5	
2 3	
5 0	
3.0 1.0	

Задача В. Миротворцы

Имя входного файла: kfor.in
Имя выходного файла: kfor.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

N миротворцев из российского корпуса KFOR десантировались в окрестности аэропорта Слатина. Точка приземления каждого миротворца задается парой целочисленных координат (x, y) . За один шаг каждый из десантников может переместиться на соседнюю целочисленную позицию вдоль оси X или Y (т.е. одна из его координат меняется на 1 по абсолютной величине). Шаги делаются по очереди, никакие два миротворца при этом не могут находиться в одной позиции одновременно. Десантники хотят выстроиться в шеренгу - линию, параллельную одной из осей координат, в которой они стояли бы в подряд идущих целочисленных позициях. Напишите программу, которая определяет минимальное суммарное число шагов, необходимое миротворцам для того, чтобы образовать шеренгу.

Идея решения: Поскольку десантники могут перемещаться только вдоль координатных осей, то задачу можно решить отдельно по x -координатам и по y -координатам, а затем убедиться в том, что процесс перемещений можно организовать так, чтобы никакие два десантника ни в какой момент времени не находились бы в одной позиции. Тогда по одной координате мы должны за минимальное число ходов собрать десантников в одну точку, а по другой - выстроить в шеренгу. Занумеруем десантников, расположенных на оси, в порядке возрастания их координаты. Назовем центром ту точку, где стоит десантник номер $Ndiv2 + 1$, если N нечетно, и любую точку между десанниками $Ndiv2$ и $Ndiv2 + 1$, если N четно. Тогда для сбора в одной точке следует выбрать любой из центров. Пусть мы выстроили десантников в шеренгу. Заметим, что их взаимное расположение не поменялось (если один десантник стоял левее другого, то после построения это сохранилось, иначе число ходов не минимально). Если вычесть из координаты каждого десантника его номер, то все они окажутся в одной точке. Следовательно, для решения задачи построения в шеренгу нужно вычесть из координаты каждого десантника его номер, собрать всех десантников в одну точку и после прибавить номера обратно. Перебрав два случая - когда десантники по оси X собираются в точку, а по оси Y строятся в шеренгу, и наоборот, - выберем из них тот, который требует меньшего числа шагов.

Формат входного файла

Первая строка входного файла содержит целое число N - количество миротворцев ($1 \leq N \leq 10000$). Каждая из последующих N строк содержит координаты десантника -

два целых числа из диапазона $[-32768, 32767]$, разделенные пробелом.

Формат выходного файла

Выведите в выходной файл искомое количество шагов.

Примеры

kfor.in	kfor.out
3 -1 -1 0 0 1 1	2

Задача C. SOS

Имя входного файла: **sos.in**
Имя выходного файла: **sos.out**
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В океане в точке с координатами (X, Y) потерпел крушение корабль. (Недалеко от места катастрофы находится остров, имеющий форму N -угольника (не обязательно выпуклого). Спасшиеся после кораблекрушения пассажиры оказались в спасательной шлюпке, которая может двигаться относительно воды в любом направлении со скоростью, не превосходящей V . В процессе движения шлюпка может менять как направление, так и величину своей скорости. В океане имеется постоянное течение, вектор скорости которого – (VT_x, VT_y) . Тем самым, вектор скорости шлюпки относительно земли определяется как сумма вектора скорости течения (VT_x, VT_y) и вектора скорости шлюпки относительно воды (V_x, V_y) . Требуется найти минимальное время, за которое шлюпка сможет добраться до острова, либо определить, что из-за сильного течения это невозможно.

Идея решения: Геометрическое место точек, где может оказаться шлюпка через время t , представляет собой расширяющийся круг радиуса Vt , плывущий по направлению течения со скоростью (VT_x, VT_y) . Нас интересует первый момент времени, в который этот круг коснется заданного N -угольника. Ясно, что он может коснуться либо его вершины, либо его стороны. Первый случай соответствует высадке людей из шлюпки в вершине острова, второй – на стороне острова. Понятно, что для минимизации времени шлюпка должна плыть с максимальной собственной скоростью V не меняя направления движения.

В случае высадки в вершине острова скорость шлюпки относительно земли должна быть направлена на данную вершину. Надо учесть, что, в общем случае, не до всех вершин можно доплыть, так как при малой собственной скорости шлюпки рассматриваемый круг может проплыть мимо них. В случае высадки на стороне острова необходимо направить собственную скорость шлюпки перпендикулярно данной стороне. Здесь возможен случай, что шлюпку снесет течением, и она пройдет мимо этой стороны острова. В таком случае вариант высадки на этой стороне не рассматривается. Тем самым, для решения задачи необходимо рассмотреть все возможные варианты высадки и выбрать тот из них, который минимизирует затрачиваемое время.

Формат входного файла

Входной файл содержит (в указанном порядке)

следующие данные: координаты (X, Y) места крушения, количество вершин острова N ($3 \leq N \leq 50$), координаты вершин острова, заданные в порядке обхода острова по часовой стрелке ($2N$ чисел), максимальную скорость спасательной шлюпки V ($V > 0$) и вектор скорости течения (VT_x, VT_y) . Все числа во входном файле, кроме N , являются вещественными и разделяются пробелами и/или символами перевода строки.

Формат выходного файла

Выведите в выходной файл искомое время не менее чем с 6 верными значащими цифрами. Если шлюпка до острова доплыть не сможет, выходной файл должен содержать сообщение `<добраться невозможно>`.

Примеры

sos.in	sos.out
4 3 3 0 0 0 3 3 0 2 1 1	4.828427