

Група А

Задача А1. КРЪГОВЕ, автор Веселин Кулев

За приближавания летен сезон, първи Европейски туристически сезон на нашето Черноморие, управата на Община Варна решила да оцвети плочките на един от площадите и възложила задачата на известен художник. Той направил идеен проект, предвиждащ да се оцветят плочките, които се съдържат в няколко кръга с еднакъв радиус. Кръговете могат да се припокриват. Една плочка се съдържа в кръг, ако има част от нея с ненулево лице, която е в кръга. За доставчик на боята художникът посочил фирмата ColXor. Оказало се, че боите на тази фирма имат много интересно свойство – когато оцветиш повторно една плочка с боята на ColXor, ефектът е, че вторият пласт боя унищожава първия и плочката става неочветена (вж. на Фигурата). Следователно плочките, които имат общо лице с два, четири и т.н. кръга, не си струва да се оцветяват и така ще се спести много боя. Напишете програма **COLXOR**, която да определя броя на плочките, които трябва да бъдат оцветени, за да се реализира идеята на художника. Ще считаме, че плочките на площада съвпадат с квадратчетата със страна единица и целочислени координати на върховете в правоъгълна координатна система.

Вход

На първия ред на стандартния вход ще бъдат зададени две цели – броят N на кръговете и радиусът им R , $1 \leq N \leq 1000$, $1 \leq R \leq 1000$. На всеки от следващите N реда ще бъдат зададени, разделени с един интервал, координатите X и Y на центъра на една от кръговете – цели числа в интервала $[-10000, 10000]$.

Изход

На единствения ред на стандартния изход програмата трябва да изведе броя на квадратчетата, които трябва да бъдат оцветени.

ПРИМЕР

Вход
2 5
3 -3
6 1

Изход
104

Решение

Под „обхождане на кръг“ ще разбираме обхождане на всички квадратчета със страна 1 от големия квадрат, описан около кръга. При това за всяко квадратче проверяваме дали се включва в кръга. Критерий за това може да бъде

наличието на поне един връх на квадратчето, който се намира в кръга (т.е. разстоянието от него до центъра на кръга да е по-малко от R). Тази процедура може да се направи за един условен кръг с радиус R , например този с център в началото на координатната система. Резултатът от обхождането може да бъде различен, в зависимост от това как ще продължи по-нататък работата на алгоритъма.

Ако знаехме, че покритите от кръговете области не се пресичат и при обхождането на кръг намерим броя C на покритите от него квадратчета, то търсеният отговор ще бъде $C \cdot N$ и това ще бъде решение със сложност $O(R^2)$. Проверката за непресичане е доста неприятна и ще вдигне сложността до $O(N^2 + R^2)$, без да имаме гаранция, че такова решение ще даде верен отговор дори за един тест.

Наивното решение, което би могло да се използва, ако нямаше ограничения по време и памет би изглеждало така. Нека A и B са страните на минималния правоъгълник, в който се съдържат всички кръгове. За всяко от квадратчетата на правоъгълника отделяме памет от един бит с начална стойност 0. Обхождаме всеки един от кръговете (но вече не условно, а според мястото му в равнината) и за всяко покрито от кръга квадратче инвертираме съответния му бит. Броят на битовете, стойността на които е 1 в края на обхождането е търсеният резултат. Сложността на този алгоритъм е $O(A \cdot B + NR^2)$, а необходимата памет $O(A \cdot B)$ – в случая около 400 мегабита. Ако вместо да отделяме по един бит за всяко квадратче използваме хеш-таблица само за квадратчетата, които се покриват от поне един кръг, тогава сложността ще падне до $O(NR^2)$, но сега пък необходимата памет може да се вдигне до $O(NR^2)$, което в случая е 1000 мегабита.

Идеята за по-добро решение изключва разглеждането на всяко квадратче. Нека при обхождането на условия кръг с център началото на координатната система разбием покриваните от кръга квадратчета на $2R$ вертикални ленти с ширина 1, като всяка лента се определя с два от върховете си, например (x, y) и $(x, y+z)$. За всеки от реалните кръгове разбиването се получава като към x и y добавим съответните координати на радиуса. Сортираме така полученото множество от $4 \cdot NR$ точки по x , а при равен x – по y . За решаването на задачата е достатъчно да обходим този масив, като за точките с еднаква x координата броим квадратчетата от първата до втората точка за оцветени, от втората до третата за неочветени, от третата до четвъртата за оцветени и т.н. Сложността на получения алгоритъм, при използване на някой от алгоритмите за бързо сортиране ще бъде $O(NR \cdot \log(NR))$.

Тава решение може да бъде подоброено. Нека, вместо да събираме кращата на всички ленти в един масив, построим отделен списък на лентите с една и съща x -координата и да сортираме потделно всеки от тези масиви. Интуитивно, най-лошият случай ще бъде когато всички кръгове имат центрове с равни x -координати. Броят на списъците в такъв случай е $2R$ а във всеки от списъците ще има по N ленти. При сортиране на всеки от списъците с някой от бързите алгоритми ще получим алгоритъм със сложност $O(NR \cdot \log N)$.