**Incinerator**

(Решение)

В задачата се искаше по даден набор от **N** незастъпващи се кръга с центрове в квадрат с размери 1000 на 1000 да се намери най-големият нов кръг (отново с център този квадрат), който може да бъде поставен без да застъпва никой от тях.

В случая, в който N = 2 може да бъде показано, че центърът на новия кръг ще лежи някъде по контура на квадрата (което е по-лесна задача).

С всяко N > 2 може да се направи тест, в който центърът на новия кръг да не лежи по контура, което е сложната версия на задачата. Реално броят кръгове от тук нататък влияе единствено на това колко ефективно трябва да е решението ви откъм скорост (с N ≤ 20 можете да си позволите и много бавни решения).

За да решим задачата трябва да знаем (или да се сетим) един що-годе стандартен трик – вместо да поставяме нов кръг с радиус R, ще увеличим радиусите на всички останали кръгове с R и ще се опитаме да поставим точка (реално, ако квадратът не е изцяло покрит от N-те кръга с уголемени радиуси, това е възможно). Разбира се, след увеличаването на радиусите на N-те кръга вече е възможно те да се застъпват (но никога един кръг не може да покрие друг изцяло).

Нека да видим как можем да проверим дали квадратът е изцяло запълнен след като сме увеличили радиусите на N-те кръга с дадено число R. Нека си представим, че квадратът не е изцяло запълнен, тоест можем да сложим точка някъде в него, която не е покрита от никой от кръговете. Нека почнем да местим тази точка наляво докато или стигнем ръба на кръг, или ръба на квадрата. След това нека почнем да я плъзгаме по ръба на кръга (или да я плъзгаме надолу, в случая, че сме стигнали до края на квадрата) докато стигнем контура на друг кръг, или долната страна на квадрата. В момента, в който стигнем някаква такава точка, спираме. Все още точката не е покрита от никой кръг (тъй като сме тръгнали от непокрита част на квадрата и единствено сме я плъзгали по контури като през цялото време тя не е била покрита). В момента, в който сме спрели, има три варианта:

1. Намираме се в пресечната точка на две страни на квадрата (реално, можем да сме само в долния ляв ъгъл на квадрата).
2. Намираме се в пресечната точка на някой от (уголемените) кръгове и някоя от страните на квадрата.
3. Намираме се в пресечната точка на два от (уголемените) кръгове.

За да решим задачата ще направим обратния процес:

1. Ще намерим всички пресечни точки на две страни на квадрата (4-те му ъгъла)
2. Ще намерим пресечните точки на всеки (уголемен) кръг с четирите страни на квадрата (най-много 8 такива за всеки кръг)
3. Ще намерим пресечните точки на всеки два от (уголемените) кръгове (най-много 2 такива за всяка двойка кръгове)

Тъй като N e до 100 получаваме най-много 4 + 800 + 10100 = 10904 пресечни точки (реално малко по-малко, но от този порядък – съвсем малко над 10000).

За всяка от тях трябва да проверим дали не лежи стриктно в някой от кръговете. В случай, че има такава, която е само по контури (но не стриктно в никой от кръговете), то квадратът не е изцяло покрит. Разбира се, тази точка трябва да е и вътре в квадрата за да е валиден отговор.

Знаем, че с R = 0 със сигурност има такава точка (няма как да поставим два или повече незастъпващи се кръга в квадрат, така че те да го покрият изцяло). С R = 1415 пък квадратът е със сигурност изцяло покрит. Следователно, отговорът трябва да е някъде между тези две числа. Всъщност, увеличавайки R започвайки от 0, квадратът не е изцяло покрит докато стигнем някакво число X (отговорът на задачата), а от там нататък е постоянно покрит. Такъв тип функции са страхотни за прилагане на двоично търсене – което трябва да направим и тук.

Така, алгоритъмът е следния:

1. Правим двоично търсене по отговора
2. За всеки радиус R, който пробваме, намираме всички пресечни точки на уголемените с R кръгове с други от тях или страните на квадрата.
3. Проверяваме дали някоя от тези пресечни точки е вътре в квадрата, но не лежи стриктно в никой от кръговете. Ако това е така, пробваме по-големи R; в противен случай, продължаваме с по-малки такива.

Сложността ни е O(log) за двоичното. Реално, тъй като е с double-и, можем да го направим с фиксиран брой итерации – трябва ни прецизност от 9 знака след десетичната точка, плюс до 4 знака преди нея, тоест 13 знака общо. Log2(1000000000000) = 40 – тоест около 40 итерации ще ни стигнат. Обикновено когато работим с двоично с double-и е по-лесно директно да правим 64 итерации за да сме сигурни (тъй като double precision числата се пазят с 64 бита, а всяка итерация на двоичното ни намира грубо по един от тях).

За всяка итерация на двоичното имаме да намерим O(N^2) пресечни точки, и за всяка от тях да проверим дали лежи в някой от кръговете за още O(N). Така сложността за цялата задача става O(log \* N^3). Откъм максимален брой операции, които можем да направим, имаме 64 \* 100 \* 100 \* 100 = 64,000,000, което минава за под секунда на модерен процесор (дори ползвайки double-и и "тежки" операции като sqrt()).

Задачата може да се реши и с по-ниска сложност, но значително по-сложно, затова избрах този вариант.

Автор: Александър Георгиев