

Eight-Shaped Figures

Идея: Геннадий Короткевич
Разработка: Геннадий Короткевич

Для решения задачи воспользуемся методом сканирующей прямой. Наша прямая будет вертикальной и идти слева направо. Разделим каждую окружность на две половины — верхнюю и нижнюю. В процессе сканирования будем поддерживать упорядоченное сверху вниз множество дуг, пересекаемых текущей прямой.

Заметим, что поскольку никакие две окружности не пересекаются по двум точкам, любые две дуги, активные на одной и той же вертикальной прямой, всегда будут упорядочены одинаково. Следовательно, нужно только научиться вставлять в множество две половины окружности i , когда сканирующая прямая достигнет координаты $x_i - r_i$, и удалять их, когда прямая достигнет координаты $x_i + r_i$.

Для эффективной обработки операций вставки и удаления в качестве упорядоченного множества следует использовать сбалансированное дерево поиска. В частности, в языке C++ можно использовать `std::set`, если самостоятельно определить компаратор для дуг.

Поскольку окружности и дуги могут касаться, требуется аккуратность в сравнении дуг между собой. Как метод сканирующей прямой поможет нам найти ответ на задачу? Заметим, что если две окружности касаются (внутренним или внешним образом), то их соответствующие дуги в какой-то момент окажутся соседними в множестве. Давайте в каждый момент, когда какие-то две дуги окажутся соседними в множестве, проверим, не касаются ли соответствующие окружности, и если касаются, сохраним их точку касания и номера окружностей, касающихся в этой точке.

После окончания процесса для каждой точки касания посчитаем, сколько окружностей касаются эту точку с одной и с другой стороны, и добавим к ответу произведение этих двух чисел.

Чтобы избежать проблем с точностью вычислений, можно хранить все точки касаний в рациональных координатах — обе координаты точек касания могут быть представлены в виде дробей с целыми числителями порядка 10^{18} и знаменателями порядка 10^9 .

Так как добавление и удаление в упорядоченное множество можно выполнить за $O(\log n)$, сложность решения составит $O(n \log n)$.

Следует обратить внимание, что, по аналогии со вторым примером, можно построить тест с ответом 10^{10} , не влезающим в 32-битный целочисленный тип (10^5 окружностей, касающихся внутренним образом в одной и той же точке слева, и 10^5 таких же окружностей справа), поэтому решения, подсчитывающие все подходящие пары окружностей по одной, недостаточно эффективны.