

Убежища

Автор задачи: Даниил Орешиников, разработчик: Павел Скобелкин

Воспользуемся бинарным поиском по ответу D — максимальному расстоянию до ближайшего кокона. Для фиксированного D проверим, хватает ли имеющихся r коконов плюс не более $k - r$ новых, чтобы покрыть всех жителей.

Проверять будем жадно. Отсортируем $x_1 \leq \dots \leq x_n$ и $a_1 \leq \dots \leq a_r$. Каждый фиксированный кокон покрывает отрезок $[a_j - D, a_j + D]$. Будем перебирать x от меньших к большим и поддерживать указатель на имеющийся кокон. Если его не хватает (то есть ближайший из имеющихся коконов находится на расстоянии $> D$), то поставим новый — выгоднее всего нам поставить его в точку $a_i + D$, так как все жители левее уже покрыты, и значит имеет смысл покрыть как можно больше тех, что справа.

В конце посчитаем, сколько коконов нам пришлось добавить — если не более $k - r$, то такое значение D достижимо (причём, расстановку новых коконов мы получили), иначе нет. То есть, в зависимости от этого будем двигать или левый указатель, или правый.

Осталось определить, когда нужно останавливать бинарный поиск. Можно было решить задачу, используя тип `double` с достаточной точностью, но при внимательном рассмотрении становится понятно, что ответ всегда будет полуцелый (то есть, целое число, или целое число, деленное на 2). Таким образом, если домножить все координаты на 2, можно было обойтись целочисленным бинарным поиском.

Тогда решение работает за $\mathcal{O}(n + r)$ на одну проверку для бинарного поиска, и один запрос работает за $\mathcal{O}(\log C \cdot (n + r) + r \log r)$. Суммарное время работы решения — $\mathcal{O}(n \log n + (nq + R \log R) \log C)$, что легко укладывается в лимиты при заданных ограничениях.