
Разбор задачи «Женитьба»

Для решения задачи можно применить жадный алгоритм. Заметим, что если в исходной расстановке детей взять ближайших друг к другу мальчика и девочку и объединить их в пару, то ни у одного из них не может возникнуть соблазна, так как они являются наиболее симпатичными друг для друга. После того, как мы объединили этих детей в пару, можно заметить, что на оставшемся разбиении на пары они никак не могут сказаться, ведь, как мы уже выяснили, ни у одного из них не может возникнуть соблазн. Значит, про этих мальчика и девочку можно забыть и решать задачу независимо для всех остальных детей. Теперь среди этого множества детей выберем ближайших мальчика и девочку, и повторим весь процесс заново. После n таких операций объединения ближайших мальчика и девочки в пары, мы получим разбиение на пары, в котором не может возникнуть соблазнов.

Таким образом, мы получаем следующий алгоритм: нужно n раз взять пару из мальчика и девочки, расстояние между которыми минимально среди всех возможных пар, объединить их друг с другом и убрать из рассмотрения. Легко понять, что такой жадный алгоритм дает нам корректное разбиение на пары, а значит, ответа -1 в задаче быть не может.

Осталось лишь эффективно реализовать данный алгоритм. Для этого можно сделать следующее замечание: если расположить всех детей в порядке возрастания координаты на прямой, то парой из мальчика и девочки, расстояние между которыми минимально, обязательно будет пара, в которой эти мальчик и девочка стоят подряд. Таким образом, достаточно поддерживать структуру, в которой можно хранить упорядоченных по возрастанию координаты детей, а также удалять их из этой структуры, когда соответствующий ребенок объединяется в пару и исключается из рассмотрения. В качестве такой структуры пойдет, например, `std::set` в языке C++ или `TreeSet` в Java. В такой же структуре можно поддерживать отрезки между соседними детьми разного пола, храня их в порядке возрастания длины. Тогда чтобы выбрать очередную пару, надо лишь взять минимальный по длине отрезок из этой структуры, а при удалении детей из рассмотрения нужно удалять старые отрезки, которые они образовывали, и добавить новые. Легко заметить, что таких отрезков всегда будет константное число.

Таким образом, если запросы ко всем структурам работают за время $\mathcal{O}(\log n)$, то асимптотика времени работы решения составит $\mathcal{O}(n \log n)$.