

Day Streak

Идея: Геннадий Короткевич
Разработка: Геннадий Короткевич

Изначально мы находимся в часовом поясе $t = 0$. Давайте поддерживать значения $d_i = \lfloor \frac{a_i+t}{m} \rfloor$ — дни, в которые решены задачи. Начнём увеличивать t вплоть до $m - 1$. Тогда в процессе каждое d_i не более одного раза увеличится на 1.

Будем поддерживать два упорядоченных множества. Множество B будет содержать все такие позиции i , что $d_{i+1} - d_i > 1$ — то есть все такие позиции, где обрывается цепочка подряд идущих дней с решёнными задачами. Множество S будет содержать длины всех имеющихся цепочек подряд идущих дней.

Тогда при увеличении d_i на единицу достаточно сделать $O(1)$ операций с множествами B и S , чтобы поддержать их в актуальном состоянии. В качестве структуры для множества можно использовать, например, `std::set` в C++ и его аналоги в других языках программирования.

Временная сложность решения составит $O(n \log n)$.

Альтернативно, можно использовать метод двух указателей, если научиться отвечать на вопрос «могут ли задачи $i, i + 1, \dots, j$ быть решены без пропусков дней». Для ответа на вопрос можно использовать дерево отрезков по часовым поясам, которое для каждого часового пояса будет хранить то, сколько пар задач $(k, k + 1)$ образуют пропуск в этом поясе ($i \leq k < j$). Тогда, если минимум по всему дереву отрезков равен 0, то существует часовой пояс, в котором пропусков нет, и можно двигать правый указатель, иначе нужно двигать левый указатель. Сложность такого решения также составит $O(n \log n)$.