

Разбор задачи «Е. Субботник»

Автор задачи и разбора — М. Густокашин

В версии задачи для лиги В достаточно отсортировать людей по росту и разбить их на группы по C человек. В каждой группе необходимо подсчитать разницу между самым высоким и самым низким человеком и выбрать максимум для всех групп.

При решении задачи в версии для лиги А полезно сначала научиться решать следующую задачу: по известному максимальному числу неудобства необходимо определить, сколько бригад можно сформировать с таким ограничением.

Эта задача решается следующим образом: отсортируем массив по неубыванию элементов (для этого можно было использовать любой метод, работающий за $O(N \log N)$). Для удобства назовём этот массив G . Людей в бригады будем набирать с использованием «жадного» алгоритма, т.е. если человека можно записать в бригаду, то будем записывать его в эту бригаду. Несложно показать, что пропуск какого-либо человека и замена его более высоким никогда не улучшит решение (но может ухудшить).

Затем введём два указателя `start` и `end` (это указатели на элементы отсортированного массива — обыкновенные целые числа, индексы элементов), которые будут указывать на самого низкого и самого высокого человека в формируемой бригаде. Вначале, указатели `start` и `end` указывают на человека с номером 1. Если разница роста следующего человека (с номером `end+1`) и первого человека в формируемой бригаде не превышает максимального допустимого числа неудобства — добавляем человека с этим номером в бригаду (т.е. увеличиваем `end` на 1). Если же разница роста превысила максимально допустимое значение числа неудобства — будем увеличивать номер самого низкого человека в бригаде на 1 до тех пор, пока $G[\text{end}] - G[\text{start}]$ превышает максимальное число неудобства. Если в какой-то момент при выполнении этих операций количество людей в формируемой бригаде стало равно C (количество людей в укомплектованной бригаде), то увеличиваем счётчик сформированных бригад на 1 и продолжаем формирование бригад со следующего человека (т.е. присваиваем переменным `start` и `end` значение `end+1`). Такая функция за $O(N)$ операций сможет подсчитать количество бригад, которые можно сформировать при заданном максимальном числе неудобства.

Для решения задачи осталось написать всего лишь несколько строк — реализацию бинарного поиска по ответу. Суть его заключается в следующем: обозначим за `left` и `right` границы области поиска ответа (максимального числа неудобства). В начале `left` равно 0 (это нужно для случая, когда все люди имеют одинаковый рост), а `right` — росту самого высокого человека минус рост самого низкого человека (обозначим его за `b`, такая граница нужна в случае, если необходимо отправить всех людей в одну бригаду).

Возьмём среднее значение между `left` и `right`: $m = (\text{left} + \text{right})/2$. Подсчитаем количество бригад, которое можно сформировать с максимальным числом неудобства, равным m . Если количество бригад оказалось меньше, чем R , то, логично, что при меньшем числе неудобства бригад получится ещё меньше и искать стоит только между числами $m+1$ и `right`. Т.е. левую границу поиска необходимо заменить на число $m+1$. В противном случае (если число сформированных бригад оказалось больше либо равным R) заменяем правую границу поиска на m . Повторять эти действия необходимо до тех пор, пока `left` не станет равным `right` (т.е. наш поиск сойдётся на одном числе). Это число и является ответом на задачу.

Общая сложность алгоритма составит $O(N \log N) + O(N \log b)$, где b — разница между самым высоким и самым низким человеком.