

Открытая городская олимпиада по программированию для обучающихся 7–8 классов
Ханты-Мансийск, пятница, 9 марта 2018 года

Разбор задач Олимпиады

Условия задач, представленный разбор и авторские решения появятся в течение нескольких дней после окончания олимпиады на странице <https://upml.tech/events/1/> и в GitHub-аккаунте <https://github.com/hminfclub/>, а также будут отправлены на электронную почту участникам.

Задача А. Успешный абитуриент

Автор: Арслан Акназаров

Данная задача проверяла умение использовать условные операторы.

Необходимо с помощью условного оператора выбрать один из четырех случаев: $a \geq 404$, $322 \leq a < 404$, $239 \leq a < 322$, $a < 239$, и вывести соответствующую строку.

Задача В. Не решай, подумой

Автор: Никита Сычев

Примечание. Опечатка в названии была допущена намеренно, данное название является пародией на известный интернет-мем.

Данная задача была размещена в наборе для разминки участников. Необходимо было из длинного текста извлечь короткую суть задачи.

Очевидно, что ряд натуральных чисел и так «отсортирован по возрасту». Поэтому нам нужно вывести k -тое натуральное число. Очевидно, что это и есть k .

Задача С. Делимость

Автор: Арслан Акназаров

Данная задача предполагала знание циклов.

Здесь просто необходимо реализовать то, что требуется — пройти по числам от 1 до n и посчитать те, на которые k не делится нацело (остаток от деления не равен нулю).

Можно было получить до 84 баллов, если перебирать делители k , и проверять, что они меньше или равны n . Также 84 балла набирали решения, в которых используются 32-битные типы вместо 64-битных.

Задача D. Подарки

Автор: Никита Сычев

Данная задача предполагала знание массивов.

Для начала научимся решать эту задачу «медленно». Считаем все числа в массив, затем переберем левую границу отрезка, затем правую (вложенным циклом). Проверим в каждом отрезке сумму, и, если она равна A , прибавим единицу к ответу.

Данное решение набирает 20 баллов (при очень хорошей реализации — 56). Поймем, насколько оно медленно: мы делаем тройной вложенный цикл, а значит, время работы пропорционально n^3 . За секунду компьютер может выполнить около 10^9 операций.

Уберем один цикл: заметим, что сумма на отрезке $[l, r + 1]$ равна сумме на отрезке $[l, r]$ плюс элементу a_{r+1} , значит можно пересчитывать сумму одновременно со сдвигом правой границы. Данное решение уже делает число операций, пропорциональное n^2 и гарантированно набирает 56 баллов.

Другой способ убрать цикл — использовать префиксные суммы. Заметим, что если $p[i]$ — сумма на отрезке $[1, i]$, то сумма на отрезке $[l, r]$ равна $(p[r] - p[l - 1])$.

Вся идея задачи заключалась в том, что необходимо придумать способ убрать ещё один цикл.

Есть два способа решения этой задачи.

Простой способ (и наиболее быстрый) основывается на принципе «двух указателей». Заметим, что если один отрезок «вложен» в другой, то сумма во «внешнем» отрезке **строго больше**, чем во внутреннем.

Тогда поступим следующим образом: зафиксируем левую границу на первом элементе и будем перебирать правую границу, пока сумма внутри отрезка не станет хотя бы A . Если она стала равна A , прибавим единицу к ответу. Далее заметим, что первый элемент не может лежать ни в каких других отрезках (иначе текущий отрезок был бы вложен в него, а в нём уже сумма хотя бы A).

Значит, мы можем сдвинуть левую границу вправо. Но правую границу не нужно перебирать заново — если её двигать левее, то мы получим отрезок, вложенный в рассмотренный ранее — а значит, сумма на нем станет меньше A . Тогда можно начать снова двигать вправо границу с текущей позиции.

Так мы пройдем весь массив, и найдем число отрезков.

Почему это работает быстро? Потому что на каждом шаге мы передвигаем одну из границ вправо. Каждая граница суммарно была передвинута n раз. Значит, всего было $2n$ шагов.

Сложный способ (и более медленный) заключается в применении бинарного поиска. Переберем левую границу l и зафиксируем её на каждом шаге.

Тогда сумма на отрезке $[l, r]$ строго возрастает с увеличением r , а, следовательно, можно применить бинпоиск, чтобы найти такое r , что сумма на отрезке будет равна A (или узнать, что такого нет).

Время работы такого решения пропорционально $n \cdot \log n$.

От 56 до 77 баллов получали решения, использующие 32-битные типы, 100 баллов получали решения, использующие 64-битные типы данных.

Задача Е. Поиск

Автор: Арслан Акназаров

Данная задача предполагала наличие базовых навыков работы со строками.

При должном знании строк (умение построить циклический сдвиг и сравнить две строки) задача требовала банальной реализации требуемого в условии.

Таким образом, сначала нам нужно перебрать циклический сдвиг: для этого переберем количество символов, которые мы обрезаем с конца. Затем построим новую строку и посимвольно сравним её с шаблоном (подходит, если на всех не совпавших позициях в шаблоне стоят знаки вопроса). Если подходит, прибавим к счетчику единицу.

Время работы решения пропорционально n^2 .

Задача F. Ядерная вакханалия

Автор: Никита Сычев

Данная задача предполагала знание базовых алгоритмов на массивах (поиск максимума).

Сначала необходимо посчитать число жителей в каждом городе после всех взрывов. Это делается просто — для каждого взрыва прибавим в соответствующую страну ещё столько же жителей, сколько пострадало.

Затем необходимо пройти по массиву и посчитать три максимальных элемента. Алгоритм аналогичен поиску второго максимума, только требует хранения трёх элементов. Время работы пропорционально n .

В качестве альтернативы в современных языках программирования можно было использовать сортировку. Она работает медленнее, но проще в реализации.

От 52 до 78 баллов могли получить решения, в которых используется собственная реализация некоторой сортировки, работающей за время, пропорциональное n^2 (пузырёк, вставками, выбором).