

Задача А(6-7). Осенний урожай

Перебираем количество деревьев с большим количеством фруктов (пусть это A), от x/A до 0 или пока остаток $(x - A*i)$ не поделится на B . Первый же найденный ответ будет правильным, если $i \geq 0$.

Задача В(6-7). К-Сумма

Прежде всего отметим, что так как каждое число может достигать *longint*, то их сумма превысит *longint* может, поэтому сумму нужно хранить в переменной типа *long long (int64)*. Те, кто использовал *int*, набрал не более 70 баллов.

Далее, наивное решение (считаем сумму k чисел от каждого числа) не проходит по времени (не больше 80 баллов). Кроме того, из-за большого объема входных данных появляется необходимость ускорения ввода из потока (`ios_base::sync_with_stdio(0); cin.tie(0);`).

Полное решение подразумевает использование префиксных сумм. То есть считаем сумму первых k элементов последовательности, затем для каждого следующего элемента добавляем i -й элемент к сумме и вычитаем $(i-k)$ -й. В итоге решение имеет сложность $O(n)$.

Задача С(6-7) А(8-9). Восстановление минимальной разницы

Так как числа всего лишь трехзначные, то задачу можно решить перебором, перебирая каждую из цифр, обозначенную '*'. При этом первая цифра не может быть 0. Из всех вариантов выбираем вариант с минимальной по модулю разнице. В худшем случае может быть 1000000 вариантов, так что перебор вполне укладывается в лимит по времени.

Задача D(6-7) В(8-9). Непростые делители

Необходимо перебрать все делители числа и просуммировать те из них, которые не являются простыми числами.

Так как большинство тестов были на небольшие тесты, то решение, проверяющее на делимость на 2, 3, 5, 7 и учитывающее, что 1 не является простым числом, брало 60 баллов.

Ограничение на 1000000 даёт возможность заработать 100 баллов даже наивному решению, то есть находим все делители просто перебирая числа от 1 до n и проверяя остаток от деления n на i . Проверку на простоту каждого делителя x также можно делать наивным образом, проверяя делимость числа на все числа от 2 до $x - 1$.

Правильным решением при условии ограничения до *longint* будет проверка на делители до корня из числа. Проверка на простоту также может быть сделана за $O(\sqrt{n})$.

Задача С(8-9). Библиотека

Наивное решение (на каждый запрос перебрать все числа в интервале) даёт 70 баллов.

Из-за большого объема входных данных появляется необходимость ускорения ввода из потока (`ios_base::sync_with_stdio(0); cin.tie(0);`).

Если бы значений жанров было не настолько много, задачу можно было бы решить статистической сортировкой (сортировка подсчетом). Тем не менее нашлись участники, применяющие этот метод сортировки на ограниченном интервале, что дало до 90 баллов.

Для полного решения необходимо отсортировать входные данные и находить решение методом поиска делением пополам (дихотомия, бинарный поиск).

Задача D(8-9). Новый год и подарки

В первую очередь заметим, что можно отсортировать числа, при этом лучше всего подойдет статистическая сортировка (сортировка подсчетом). Для формирования подарков конфеты стоит выбирать жадным алгоритмом (в первую очередь используем те конфеты, которых больше).

Наилучшие решения, использующие сортировку сравнением (например, `sort`), взяли 90 баллов.

Для ускорения выбора конфет с наибольшим количеством по типу, можно использовать приоритетную очередь. То есть помещаем в приоритетную очередь все количества конфет каждого типа, и, пока в очереди есть не меньше m элементов, извлекаем из очереди m самых больших значений, уменьшаем каждое из них на 1, и помещаем те, которые больше 0, обратно в очередь.

Задача E(8-9). Размер бака имеет значение

Классическая задача на теорию графов, поиск кратчайшего пути во взвешенном графе при отсутствии ребер отрицательного веса. Может быть решена алгоритмами Дейкстры, Флойда или Беллмана-Форда.

Использование теории графов может показаться сложным, но стоит посмотреть на реализацию алгоритмов Флойда или Беллмана-Форда, и можно увидеть, что реализация алгоритма представляет собой три вложенных цикла с проверкой, что вполне по силам ученикам 8-9 класса.