

Информатика

Златопольский Дмитрий Михайлович

Кандидат технических наук, доцент.

Доцент кафедры информатики и прикладной математики

Московского городского педагогического университета;

организатор и директор Музея истории

вычислительной техники.



Методика выполнения заданий 26 и 27 демонстрационного варианта ЕГЭ по информатике

В статье описываются методики выполнения двух самых сложных заданий¹, представленных в проекте демонстрационного варианта Единого государственного экзамена по информатике и ИКТ 2021 года [1], который пройдет в компьютерной форме.

Задание 26

Условие

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов.

Известно, какой объём занимает файл каждого пользователя.

По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Входные данные.

В первой строке входного файла находятся два числа: S – размер свободного места на диске (натуральное число, не превышающее 10000) и N – количество пользователей (натуральное число, не превышающее 1000). В следующих N строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

¹ При этом весомость этих заданий составляет не 1, а 2 балла (общий максимальный первичный балл за весь экзамен – 30).

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Пример входного файла:

1004

80

30

50

40

При таких исходных данных можно сохранить файлы максимум двух пользователей. Возможные объёмы этих двух файлов 30 и 40, 30 и 50 или 40 и 50. Наибольший объём файла из перечисленных пар – 50, поэтому ответ для приведённого примера:

250

Решение

Полный перебор всех возможных вариантов размещения файлов невозможен по следующей причине.

Все варианты такие:

1) файл одного из пользователей (N вариантов);

2) файлы двух любых пользователей;

...

N) файлы всех N пользователей (один вариант).

Но так как N может составлять 999, то рассмотрение всех вариантов в общем случае будет невозможным.

Рациональный вариант решения задачи основан на следующих рассуждениях.

Пусть $N = 14$, $s = 100$, а значения размеров файлов пользователей такие:

95 11 91 52 65 57 54 88 87 58 6 10 50 55

Очевидно, что для размещения максимального количества файлов в первую очередь следует размещать на диске файлы небольших размеров. Отсортировав указанные размеры в порядке неубывания, получим:

6 10 11 50 52 54 55 57 58 65 87 88 91 95

Далее, суммируя последовательно все начальные (меньшие) размеры можно найти такой минимальный размер файла, для которого сумма всех размеров превышает предельное значение 100. Для приведенного примера это пятое число (52). Значит, на диске можно сохранить файлы максимум четырех пользователей.

Означает ли что значение 50 является максимальным для найденного максимального количества файлов? Ответ – нет. Это может быть и большее значение, при котором сумма размеров всех файлов не превышает s . Для приведенного примера таким максимальным значением является число 65.

Прежде чем рассказывать о том, как найти окончательное максимальное значение, приведем фрагмент программы на школьном алгоритмическом языке, в котором находится первое из искомых значений – максимальное количество

размещаемых файлов. В ней, кроме переменных N и s , использованы следующие основные величины:

m – массив для хранения заданных размеров файлов пользователей;

сумма – сумма размеров файлов;

k – искомое максимальное количество размещаемых на диске файлов;

номер_прохода, лев, всп – величины, необходимые для сортировки массива методом обмена (см., например, [2])².

Итак, начало программы:

алг

нач **цел** N , s , сумма, k , номер_прохода, лев, всп, **цел таб** $m[1 : 1000]$

| Чтение значения s из входного файла

...

| Чтение значения N из входного файла

...

| Чтение из входного файла значений размеров файлов пользователей
| и запись их в массив m

...

| Сортировка массива m методом обмена

нц для номер_прохода **от** 1 **до** $N - 1$

нц для лев **от** 1 **до** $N - \text{номер_прохода}$

если $m[\text{лев}] > m[\text{лев} + 1]$

то | Проводим обмен

 всп := $m[\text{лев}]$

$m[\text{лев}] := m[\text{лев} + 1]$

$m[\text{лев} + 1] := \text{всп}$

все

кц

кц

| Ищем первое искомое значение, используя оператор цикла

| с условием

$i := 0$ | Условно для проверки первого элемента массива

сумма := 0

нц пока сумма + $m[i + 1] \leq S$ | Проверяем следующий элемент

| Учитываем его

сумма := сумма + $m[i + 1]$

| и переходим к нему

$i := i + 1$

кц

| Первое искомое значение – максимальное количество файлов k

| найдено (оно равно последнему значению i)

$k := i$

| Сумма для найденного количества равна сумма

² Можно применить любой другой способ сортировки.

Теперь о том, как определить второе искомое значение (в примере данных, приведенных нами выше.)

Можно исключить из общей суммы размеров файлов значение 50 и попытаться найти среди остальных (бóльших) размеров максимальное значение, при котором общая сумма размеров не превышает значение s . Это можно сделать с использованием оператора цикла с условием:

```
сумма_без = сумма - м[к]    | Смысл величины сумма_без понятен из  
                             | ее имени
```

```
і := к | Начинаем с найденного до этого к-го элемента
```

```
нц пока сумма_без + м[і] <= s  
    | Рассматриваем следующий элемент  
    і := і + 1
```

кц

После этого возможны два варианта:

- 1) большее значение найдено (если окажется $i > k$)
- 2) большее значение не найдено (значение $i = k$ не изменилось).

В обоих случаях искомым максимальным значением будет значение $m[i]$. Например, для приведенных значений размеров это будет, как уже отмечалось, 65 ($i = 10$), а для таких размеров файлов:

6 10 11 50 87 88 91 95

– ответом будет все то же значение 50 (i останется равным 4). Такой же ответ будет для следующего случая:

6 10 11 50 50 87 88 91 94 95

($i = 5$).

...

| Поиск второго искомого значения

```
сумма_без = сумма - м[к]
```

```
і := к | Начинаем с найденного до этого к-го элемента
```

```
нц пока сумма_без + м[і + 1] <= s    | Подходит следующий элемент
```

```
    | Переходим на него
```

```
    і := і + 1
```

кц

| Вывод ответов

```
вывод к, " ", м[і]
```

| Заккрытие файла

...

кон

Задание 27

Условие

Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Гарантируется, что искомую сумму получить можно. Программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи.

Входные данные

Даны два входных файла (файл *A* и файл *B*), каждый из которых содержит в первой строке количество пар N ($1 \leq N \leq 100000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример организации исходных данных во входном файле:

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

Для указанных входных данных значением искомой суммы должно быть число 32.

В ответе укажите два числа: сначала значение искомой суммы для файла *A*, затем для файла *B*.

Предупреждение: для обработки файла *B* не следует использовать переборный алгоритм, вычисляющий сумму для всех возможных вариантов, поскольку написанная по такому алгоритму программа будет выполняться слишком долго.

Решение

Приведем решение применительно к любому из файлов.

Чтобы получить максимально возможную сумму, будем брать из каждой пары самое большое число. Если полученная при этом сумма будет делиться на 3, ее необходимо уменьшить. Для этого достаточно в одной из пар, где числа имеют разные остатки при делении на 3, заменить ранее выбранное число на другое число из той же пары. При этом разница между числами в паре должна быть минимально возможной.

Основные величины, которые будем использовать в соответствующей программе:

N – количество пар чисел;
 $a_{\text{макс}}$ – наибольшее возможное число в исходных данных (константа, равная 10000);
 a_1 – первое число в паре чисел;
 a_2 – второе число в паре чисел;
 макс – максимум в паре;
 мин – минимум в паре;
 сумма – сумма выбранных чисел;
 мин_разн – минимальная разность $\text{макс} - \text{мин}$, не кратная 3.

Программа:

алг

```
нач цел N, a1, a2, a_макс, макс, мин, сумма, мин_разн, i
a_макс := 10000
| Начальное значение величины мин_разн
мин_разн := a_макс + 1
| Чтение из первой строки файла значения количества пар
| чисел и запись его в переменную N
...
| Определяем сумму максимальных значений в парах
сумма := 0
нц для i от 1 до N
| Чтение из остальных строк файла значений пар чисел
| и запись их в переменные a1 и a2
...
если a1 > a2
то
макс := a1; мин := a2
иначе
макс := a2; мин := a1
все
сумма := сумма + макс
| При необходимости уточняем значение мин_разн
если mod(макс - мин, 3) > 0 и макс - мин < мин_разн
то
мин_разн := макс - мин
все
кц
| Проверяем сумму
если mod(сумма, 3) = 0
то | Уточняем ее в зависимости от значения мин_разн
если мин_разн > a_макс
то
сумма := 0
иначе
сумма := сумма - мин_разн
все
все
| Вывод ответа
вывод нс, сумма
кон
```

Литература

1. <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!/tab/151883967-5>.
2. Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы. – М., Бином. Лабораторий знаний, 2007.