

Практическая работа №3. Алгоритмы сортировки

Задача 1. Максимальная тройка

Ограничение по времени: 1.5 секунд

Ограничение по памяти: 8 мегабайт

Имеется не более 1000000 целых чисел, каждое из которых лежит в диапазоне от -1000000 до 1000000. Найти максимально возможное значение произведений любых трех различных по номерам элементов массива.

Формат входных данных:

N

A1

A2

...

AN

Формат выходных данных:

MaxPossibleProduct

Примеры:

Стандартный ввод	Стандартный вывод
10 -1 2 3 -4 -2 5 -1 5 -3 -2	75

Задача 2. Сортировка по многим полям

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

В базе данных хранится N записей, вида (Name, a_1 , a_2 , ..., a_k) – во всех записях одинаковое число параметров. На вход задачи подается приоритет полей – перестановка на числах 1, ..., k – записи нужно вывести по

невозрастанию в соответствии с этим приоритетом. В случае, если приоритет полей таков: 3 4 2 1, то это следует воспринимать так: приоритет значений из 3 колонки самый высокий, приоритет значений из колонки 4 ниже, приоритет значений из колонки 2 еще ниже, а приоритет значений из колонки 1 самый низкий.

Формат входных данных:

$N \leq 1000$

$k: 1 \leq k \leq 10$

$p_1 p_2 \dots p_k$ – перестановка на k числах, разделитель – пробел

N строк вида

Name $a_1 a_2 \dots a_k$

Формат выходных данных:

N строк с именами в порядке, согласно приоритету

Примеры:

Стандартный ввод	Стандартный вывод
3	B
3	A
2 1 3	C
A 1 2 3	
B 3 2 1	
C 3 1 2	

Замечание. Так как колонка под номером 2 самая приоритетная, то переставить записи можно только двумя способами: (A, B, C) и (B, A, C). Следующий по приоритетности столбец – первый, и он позволяет выбрать из возможных перестановок только (B, A, C). Так как осталась ровно одна перестановка, третий приоритет не имеет значения.

Задача 3. Оболочка.

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Имеется массив из N целочисленных точек на плоскости.

Требуется найти периметр наименьшего охватывающего многоугольника, содержащего все точки.

Формат входных данных:

N

x1 y1

x2 y2

...

xn yn

$$5 \leq 500000$$

$$-10000 \leq x_i, y_i \leq 10000$$

Формат выходных данных:

Одно вещественное число – периметр требуемого многоугольника с двумя знаками после запятой.

Примеры:

Стандартный ввод	Стандартный вывод
5 2 1 2 2 2 3 3 2 1 2	5.66

Задача 4. Очень быстрая сортировка.

Ограничение по времени: 1.5 секунд

Ограничение по памяти: 512 мегабайт

Имеется рекуррентная последовательность A_1, A_2, \dots, A_N , строящаяся по следующему правилу:

$$A_1 = K$$

$$A_{i+1} = (A_i \times M) \% (2^{32} - 1) \% L$$

Требуется найти сумму всех нечетных по порядку элементов в отсортированной по неубыванию последовательности по модулю L .

Для входных данных

5 7 13 100

последовательность будет такой:

$\{7; 7 \times 13\%100 = 91; 91 \times 13\%100 = 83; 83 \times 13\%100 = 79; 79 \times 13\%100 = 27\}$, то есть $\{10; 91; 83; 79; 27\}$.

Отсортированная последовательность $\{7; 27; 79; 83; 91\}$.

Сумма элементов на нечетных местах $= (7 + 79 + 91)\%100 = 77$.

Формат входных данных:

N K M L

$5000000 \leq N \leq 60000000, 0 \leq K, L, M \leq 2^{32} - 1$

Формат выходных данных:

RESULT

Примеры:

Стандартный ввод	Стандартный вывод
5 7 13 100	77

Замечание. Для представления элементов последовательности необходимо использовать тип данных `unsigned int`.

Для получения массива используйте цикл:

`a[0] = K;`

`for (int i = 0; i < N-1; i++)`

`a[i+1] = (unsigned int) ((a[i]*unsigned long long)M)&0xFFFFFFFFU)%L;`

Внимание! Изменение типа данных и/или метода генерации элементов массива может привести (и на различных компиляторах приводит) к другой последовательности!

Задача 5. Внешняя сортировка.

Ограничение по времени: 2 секунды

Ограничение по памяти: 2 мегабайта

В файле «input.txt» содержатся строки символов, длина каждой строки не превышает 10000 байт. Файл нужно отсортировать в лексикографическом порядке и вывести результат в файл «output.txt». Вот беда, файл занимает много мегабайт, а в Вашем распоряжении оказывается вычислительная система с очень маленькой оперативной памятью. Но файл должен быть отсортирован!

Примеры:

input.txt	output.txt
qwertyuiopasdffghhj qpoiuytredgfhfd asdfghjklvcvx alkjghcdysdfgsr pquytrgsdjdsa akjhfgdghshhfuushvdfs	akjhfgdghshhfuushvdfs alkjghcdysdfgsr asdfghjklvcvx pquytrgsdjdsa qpoiuytredgfhfd qwertyuiopasdffghhj