

## Задача А. Планирование заданий

Имя входного файла: `schedule.in`  
Имя выходного файла: `schedule.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Имеется некоторое множество заданий и один исполнитель. На выполнение одного задания уходит единица времени. Задания можно выполнять начиная с момента времени 0. У каждого задания есть две характеристики:  $d_i$  и  $w_i$ . Если задание не было выполнено к моменту времени  $d_i$ , взимается штраф в размере  $w_i$ . Требуется минимизировать суммарный штраф.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество заданий ( $1 \leq n \leq 100\,000$ ). Следующие  $n$  строк содержат по два натуральных числа, разделенных пробелом —  $d_i$  и  $w_i$  ( $0 \leq d_i, w_i \leq 10^9$ ).

### Формат выходного файла

Выведите одно число — минимальный суммарный штраф.

### Примеры

| <code>schedule.in</code> | <code>schedule.out</code> |
|--------------------------|---------------------------|
| 2                        | 1                         |
| 1 1                      |                           |
| 1 2                      |                           |

## Задача В. Уничтожение графа

Имя входного файла: `destroy.in`  
Имя выходного файла: `destroy.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан связный взвешенный граф. Требуется уничтожить максимальное количество ребер, чтобы были выполнены следующие условия: суммарная стоимость уничтоженных ребер не превосходит  $s$ , оставшийся после уничтожения граф должен быть связан.

### Формат входного файла

Первая строка входного файла содержит числа  $n$  и  $m$  — количество вершин и ребер в графе, и  $s$  — максимальную суммарную стоимость уничтоженных ребер ( $2 \leq n \leq 50\,000$ ,  $1 \leq m \leq 100\,000$ ,  $0 \leq s \leq 10^{18}$ ). Следующие  $m$  строк описывают ребра — для каждого ребра указаны номера вершин, которые оно соединяет, и стоимость уничтожения этого ребра (не превышает  $10^{18}$ ).

### Формат выходного файла

На первой строке выходного файла выведите максимальное количество ребер, которые можно уничтожить. На второй строке выведите их номера в порядке возрастания (ребра нумеруются с единицы в порядке, в котором они заданы во входном файле).

### Пример

| <code>destroy.in</code> | <code>destroy.out</code> |
|-------------------------|--------------------------|
| 6 7 10                  | 2                        |
| 1 2 3                   | 1 5                      |
| 1 3 3                   |                          |
| 2 3 3                   |                          |
| 3 4 1                   |                          |
| 4 5 5                   |                          |
| 5 6 4                   |                          |
| 4 6 5                   |                          |

## Задача С. Паросочетание максимального веса

Имя входного файла: `matching.in`  
Имя выходного файла: `matching.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан двудольный граф. Количество вершин в левой и правой доле совпадает и равно  $n$ . У каждой вершины левой доли есть вес,  $i$ -й вершине соответствует вес  $w_i$ . Вес паросочетания, ребрам которого инцидентны вершины левой доли  $a_1, a_2, \dots, a_k$  есть  $\sqrt{\sum_{i=1}^k w_{a_i}^2}$ . Требуется найти паросочетание максимального веса.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество вершин в обеих долях ( $1 \leq n \leq 1000$ ). Вторая строка входного файла содержит  $n$  целых чисел  $w_1, w_2, \dots, w_n$  ( $1 \leq w_i \leq 1000$ ). Следующие  $n$  строк содержат описания ребер, инцидентных соответствующей вершине левой доли. Формат описания: количество ребер, затем номера вершин правой доли, разделенные пробелом. Суммарное количество ребер не превосходит 200000.

### Формат выходного файла

Выведите  $n$  чисел — для каждой вершины левой доли выведите номер вершины правой доли, с которой ее надо взять в паросочетание. Если вершина не входит в паросочетание, выведите 0.

### Примеры

| <code>matching.in</code> | <code>matching.out</code> |
|--------------------------|---------------------------|
| 4                        | 2 1 0 4                   |
| 1 3 2 4                  |                           |
| 4 1 2 3 4                |                           |
| 2 1 4                    |                           |
| 2 1 4                    |                           |
| 2 1 4                    |                           |

## Задача D. Проверка

Имя входного файла: `check.in`  
Имя выходного файла: `check.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано некоторое семейство множеств  $S \subset 2^X$ . Требуется проверить, может ли  $S$  быть семейством независимых множеств некоторого матроида.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — мощность множеств  $X$  и  $S$  соответственно ( $1 \leq n \leq 10$ ,  $0 \leq m \leq 2^n$ ). Каждая из следующих  $m$  строк содержит описание элемента множества  $S$ . Формат описания: количество элементов в подмножестве, затем через пробел номера этих элементов. Элементы множества  $X$  занумерованы начиная с единицы.

### Формат выходного файла

Выведите «YES», если  $S$  может быть семейством независимых множеств некоторого матроида и «NO» иначе.

### Примеры

| <code>check.in</code>           | <code>check.out</code> |
|---------------------------------|------------------------|
| 2 4<br>0<br>1 1<br>1 2<br>2 1 2 | YES                    |
| 2 3<br>0<br>1 1<br>2 1 2        | NO                     |

## Задача Е. Циклы

Имя входного файла: `cycles.in`  
Имя выходного файла: `cycles.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано некоторое семейство множеств  $S \subset 2^X$ . Известно, что это множество циклов некоторого матроида. Кроме того, у каждого элемента множества  $X$  есть свой вес. Вес подмножества  $X$  есть сумма весов элементов, принадлежащих ему. Требуется найти базу максимального веса.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — мощность множеств  $X$  и  $S$  соответственно ( $1 \leq n \leq 20$ ). Вторая строка входного файла содержит  $n$  чисел  $w_1, w_2, \dots, w_n$  ( $1 \leq w_i \leq 1000$ ). Здесь элементы множества  $X$  занумерованы начиная с единицы и  $w_i$  — вес  $i$ -го элемента множества  $X$ . Каждая из следующих  $m$  строк содержит описание элемента множества  $S$ . Формат описания: количество элементов в подмножестве, затем через пробел номера этих элементов.

### Формат выходного файла

Выведите одно число — вес максимальной базы.

### Примеры

| <code>cycles.in</code>     | <code>cycles.out</code> |
|----------------------------|-------------------------|
| 3 1<br>10 20 30<br>3 1 3 2 | 50                      |