# Problem: Train Composition

The Institute of Railway Transport has assembled a number of cargo wagon prototypes in its depot a now wants to connect them to form a train pulled by one locomotive to perform a test drive. Due to various technical constraints it is not possible to connect the wagons in arbitrary order and consequently we may not even be allowed to compose a train from all the wagons. Wagon $v$ is said to be the predecessor of wagon $w$ if wagon $w$ can be put immediately behind wagon $v$. If a wagon can be placed directly behind the locomotive, then the locomotive counts as a possible predecessor of the wagon. For every wagon, we are given a list of its predecessors. Individual wagons have different weight and our main priority is that the weight of the train (excluding the locomotive) is the maximum possible.

The task is to compose a train of the maximum weight within given constraints.

## Input

For the sake of simplicity, the wagons are identified by integers 1 to $N$, the locomotive has an id of 0.
The first line of input contains a positive number $N$ representing the number of wagons in the depot. Following are $N$ lines, where each line specifies an individual wagon. Lines (wagons) can be given in arbitrary order. The specification of a wagon contains the wagon id followed by an integer that represents its weight. Next comes an unsorted list of ids of all wagon predecessors. All of this information is separated by a space character.

## Output

Output consists of two lines. The first line contains the wagon ids of the found train ordered from the beginning to the end, including the locomotive (the line starts with 0). The ids are separated by a space character. The second line gives the weight of the found train. If two or more trains have the same weight, the output only contains the train whose identification is the lowest when sorted lexicographically according to the wagon numbers. For example, from the trains 0 3 6, 0 3 1 4 2, 0 3 1 5 the train 0 3 1 4 2 would be picked (see example 4 below).

## Note

Maximum number of wagons in the depot is 30.

## Example 1

input:

```
5
1 50 0 4
2 30 1 5
3 100 4 0
4 40 0 3
5 120 4
```

output:

```
0 3 4 5 2
290
```

## Example 2

input:

```
4
1 100 2 3 4
2 100 1 3 4
3 100 1 2 4
4 100 1 2 3
```

output:

```
0
0
```

## Example 3

input:

```
5
1 10 0
2 10 1 3 4
3 10 1 2 4
4 10 1 2 3
5 90 0
```

output:

```
0 5
90
```

## Example 4

input:

```
6
3 80 0
2 20 3 4
4 30 1
1 10 3
6 60 3
5 50 1
```

output:

```
0 3 1 4 2
140
```

## Example 5

input:

```
20
1 90 11 16 19
2 52 3 7 9 12 15 17
3 97 2 5 6 9 16 20
4 85 0 3 7 13
5 74 11
6 65 0 4 9 14 16
7 89 1 5 16
8 72 3
9 71 12 14 20
10 64 5 6 12 16 17
11 97 0 15
12 82 2 6 9 13 15 19
13 52 5 11 12 15 18
14 85 1 6 10 19
15 87 0 1 6
16 63 6 7 20
17 92 8 10 13 16 17 19
18 81 6 9 14
19 83 3
20 72 7 10 11 19
```

output:

```
0 4 6 3 19 1 15 11 5 7 20 16 10 14 9 18 13 17 2 12
1481
```