Sluggish Decisional Knapsack X94664_en

We consider the classical Knapsack problem in its 0/1 variant; this means that each object is either taken fully into the knapsack, exactly once, or left fully out. This problem refers to a decisional version where both a weight limit and a desired value are specified, and where we wish to obtain all the possible solutions. Alternatively, problem X59240 corresponds instead to the more standard optimization version.
More precisely, given are a desired total value *totv*, a weight limit *limw*, and a sequence of n objects consisting of weight and value of each object. A solution is a subset of objects whose sum of weights is at most *limw* and whose sum of values is at least *totv*.
Write a program that reads an instance consisting of *totv*, *limw*, n, and the n pairs of weight and value of the n objects (in this order) and prints all the solutions: sets of object numbers where the total weight does not exceed *limw* and the total value is at least *totv*. Note that there may be instances where the answer is empty, when no solution at all is possible.

Input

Input is an instance that starts with *totv*, the minimum total value desired for the knapsack, followed by *limw*, the maximum weight, and n, the quantity of objects. Then follow n pairs: the weight and value of each of the n objects. Here n is a non-negative integer, and all the other values are positive integers.

Output

Print one line for each possible solution. In each solution, objects are identified by the numbers 0 to $n-1$, in the same order in which their weights and values were read; hence, each line must consist of the indices (between 0 and $n-1$) of the objects taken for that solution, separated by single blank spaces.

Sample input 1

```
5000
10
3
7 3000
8 6000
3 2000
```

Sample output 1

```
1
0 2
```

Sample input 2

```
25
3
3
1 10
1 10
2 20
```

Sample output 2

```
0 2
1 2
```

Sample input 3

```
45
26
```

```
5
9 16
8 15
12 24
```

```
11 23
7 13
```

Sample input 4

```
270
165
10
23 92
31 57
29 49
44 68
53 60
38 43
63 67
85 84
89 87
82 72
```

Sample output 3

```
2 3
1 3 4
```

Sample output 4

```
0 1 2 9
0 1 3 4
0 1 3 6
0 2 3 6
0 1 2 3 5
```

Observation

The lines corresponding to the solutions can be printed in any order. Further, the object indices making up each solution can be printed within the corresponding line also in any order. On the other hand, the time allowance of this problem is rather mild, and even quite inefficient solutions may be accepted (hence the "sluggish" adjective).

Problem information

Author: José Luis Balcázar

Generation: 2026-01-25T17:26:09.305Z