Greedy Practice Problems

A. Dijkstra

15 s., 1024 MB

There are n cities and m flight connections between them. Your task is to determine the length of the shortest route from city 1 to every city, using Dijkstra's algorithm.

Additionally, you need to print all cities that have a shortest distance from city 1 less than or equal to a given value x, in $ascending\ order$.

Input
The first input line has two integers n, m, and x: the number of cities, flight connections, and the maximum distance.

The cities are numbered $1,2,...,n$.

After that, there are m lines describing the flight connections. Each line has three integers a, b and c: a flight begins at city a, ends at city b, and its length is c. Each flight is a one-way flight.

You can assume that it is possible to travel from city 1 to all other cities.

Output
Print all cities (in $ascending\ order$) that have a shortest distance from city 1 that is less than or equal to x, each on a new line.

input
3 4 3 1 2 6 1 3 2 3 2 3 1 3 4
output
1 3

B. Activity Selection

5 s., 256 MB

Given an array of $intervals$ where $intervals_i$ = [$start_i$, end_i], return the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.

Note that intervals which only touch at a point are non-overlapping. For example, [1, 2] and [2, 3] are non-overlapping.

Input
The first input line has an integer n: the number of $intervals$.

After this, there are n lines that describes each interval. Each line has two integers $start_i$ and end_i: the starting and ending times of an interval.

$Constraints$

$(1 \leq n \leq 2 * 10^5, 1 \leq start_i < end_i \leq 10^9)$

Output
Print one integer: the minimum number of intervals you need to remove so that there are no overlapping intervals.

input
4 1 2 2 3 3 4 1 3
output
1

input
3 1 2 1 2 1 2
output
2

input
2 1 2 2 3
output
0

C. Fractional Knapsack

5 seconds, 256 megabytes

You are given N items, each item has a value of p_i per kilogram and is available in k_i kilograms.

You also have a knapsack that can carry a maximum weight of W.

The goal is to maximize the total value of items placed into the knapsack.

Unlike the traditional knapsack problem, you can take fractions of an item, i.e., you can divide an item into smaller parts if needed.

Input
The first input line contains two integers n and x: the number of items and the maximum weight capacity of the knapsack in kilograms.

The next line contains n integers $w_1,w_2,...,w_n$: the number of available kilograms for each item.

The last line contains n integers $v_1,v_2,...,v_n$: the value of 1 kg of each item.

$1 \leq n \leq 10^5$.

$1 \leq x \leq 10^5$.

$1 \leq h_i, s_i \leq 10^5$.

Output
Print one integer, The maximum value that can be carried in the knapsack.

input
3 10 1 2 3 10 20 30

output

```
140
```

input

```
1 100
5
10
```

output

```
50
```

input

```
3 50
10 20 30
100 2 1
```

output

```
1060
```