

## JS-组装最大可靠性设备-一个设备由 N 种类型元器件组成

题目描述：一个设备由 N 种类型元器件组成（每种类型元器件只需要一个，类型 type 编号从 0~N-1），每个元器件均有可靠性属性 reliability，可靠性越高的器件其价格 price 越贵。而设备的可靠性由组成设备的所有器件中可靠性最低的器件决定。

给定预算 S，购买 N 种元器件（每种类型元器件都需要购买一个），在不超过预算的情况下，请给出能够组成的设备的最大可靠性。

输入描述：S N // S 总的预算，N 元器件的种类

total // 元器件的总数，每种型号的元器件可以有多种；此后有 total 行具体器件的数据

type reliability price // type 整数类型，代表元器件的类型编号从 0 ~ N-1；reliability 整数类型，代表元器件的可靠性；price 整数类型，代表元器件的价格

输出描述：符合预算的设备的最大可靠性，如果预算无法买齐 N 种器件，则返回 -1

补充说明：0 <= S, price <= 10000000;

0 <= N <= 100;

0 <= type <= N-1;

0 <= total <= 100000;

0 < reliability <= 100000;

### 示例1

输入：500 3

6

0 80 100

0 90 200

1 50 50

1 70 210

2 50 100

2 60 150

输出：60

说明：预算500，设备需要3种元件组成，方案 类型0的第一个(可靠性80)，类型1的第二个(可靠性70)，类型2的第二个(可靠性60) 可以使设备的可靠性最大 60

### 示例2

输入：100 1

1

0 90 200

输出：-1

说明：组成设备需要1个元件，但是元件价格大于预算，因此无法组成设备，返回-1

```

1  const rl = require("readline").createInterface({ input: process.stdin });
2  var iter = rl[Symbol.asyncIterator]();
3  const readline = async () => (await iter.next()).value;
4
5  void async function () {
6    // Write your code here
7    let tokens = [];
8    while(line = await readline()){
9      tokens.push(line.split(" ").map(Number));
10   }
11   const [s, n] = tokens[0];
12   const [total] = tokens[1];
13   const devs = tokens.slice(2);
14
15   function checkReliability(mid, s, n, devs) {
16     let min = new Array(n).fill(10000001), cost = 0;
17     // console.log(min);
18     for (let i = 0; i < devs.length; i++) {
19       const [type, rel, price] = devs[i];
20       if (rel >= mid) {
21         min[type] = Math.min(min[type], price)
22       }
23     }
24     for(let i = 0; i < n; i++) {
25       // console.log(min[i]);
26       cost += parseInt(min[i]);
27     }
28     // console.log(min);
29     // console.log(cost);
30     return cost <= s;
31   }
32
33   let left = 1, right = 100000, res = -1;
34   while(left <= right) {
35     let mid = Math.floor((left + right) / 2);
36     if(checkReliability(mid, s, n, devs)) {
37       res = mid;
38       left = mid + 1;
39     } else {
40       right = mid - 1;
41     }
42   }
43   console.log(res);
44 }

```