CCF CSP 计算机软件能力认证

CCF CSP

第 29 次

时间：2023 年 3 月 19 日 13:30 ~ 17:30

题目名称	田地丈量	垦田计划	LDAP	星际网络 II	施肥
题目类型	传统型	传统型	传统型	传统型	传统型
输入	标准输入	标准输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	1.0 秒	12.0 秒	2.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB	1024 MiB
子任务数目	10	20	10	20	20
测试点是否等分	是	是	是	是	是

田地丈量（farm1）

【题目描述】

西西艾弗岛上散落着 n 块田地。每块田地可视为平面直角坐标系下的一块矩形区域，由左下角坐标 (x_1, y_1) 和右上角坐标 (x_2, y_2) 唯一确定，且满足 $x_1 < x_2$、$y_1 < y_2$。这 n 块田地中，任意两块的交集面积均为 0，仅边界处可能有所重叠。

最近，顿顿想要在南山脚下开垦出一块面积为 $a \times b$ 矩形田地，其左下角坐标为 $(0, 0)$、右上角坐标为 (a, b)。试计算顿顿选定区域内已经存在的田地面积。

【输入格式】

从标准输入读入数据。

输入共 $n + 1$ 行。

输入的第一行包含空格分隔的三个正整数 n、a 和 b，分别表示西西艾弗岛上田地块数和顿顿选定区域的右上角坐标。

接下来 n 行，每行包含空格分隔的四个整数 x_1、y_1、x_2 和 y_2，表示一块田地的位置。

【输出格式】

输出到标准输出。

输出一个整数，表示顿顿选定区域内的田地面积。

【样例输入】

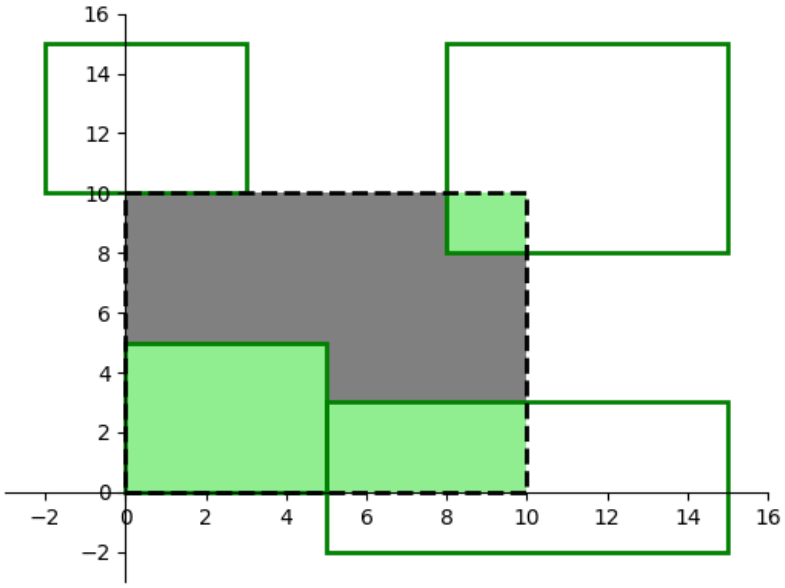
```
4 10 10
0 0 5 5
5 -2 15 3
8 8 15 15
-2 10 3 15
```

【样例输出】

```
44
```

【样例解释】

如图所示，选定区域内田地（绿色区域）面积为 44。



【子任务】

全部的测试数据满足 $n \leq 100$，且所有输入坐标的绝对值均不超过 10^4。

垦田计划（farm2）

【题目描述】

顿顿总共选中了 n 块区域准备开垦田地，由于各块区域大小不一，开垦所需时间也不尽相同。据估算，其中第 i 块（$1 \leq i \leq n$）区域的开垦耗时为 t_i 天。这 n 块区域可以同时开垦，所以总耗时 t_{Total} 取决于耗时最长的区域，即：

$$t_{Total} = \max\{t_1, t_2, \cdots, t_n\}$$

为了加快开垦进度，顿顿准备在部分区域投入额外资源来缩短开垦时间。具体来说：

- 在第 i 块区域每投入 c_i 单位资源，便可将其开垦耗时缩短 1 天；
- 耗时缩短天数以整数记，即第 i 块区域投入资源数量必须是 c_i 的整数倍；
- 在第 i 块区域最多可投入 $c_i \times (t_i - k)$ 单位资源，将其开垦耗时缩短为 k 天；
- 这里的 k 表示开垦一块区域的最少天数，满足 $0 < k \leq \min\{t_1, t_2, \cdots, t_n\}$；换言之，如果无限制地投入资源，所有区域都可以用 k 天完成开垦。

现在顿顿手中共有 m 单位资源可供使用，试计算开垦 n 块区域最少需要多少天？

【输入格式】

从标准输入读入数据。

输入共 $n+1$ 行。

输入的第一行包含空格分隔的三个正整数 n、m 和 k，分别表示待开垦的区域总数、顿顿手上的资源数量和每块区域的最少开垦天数。

接下来 n 行，每行包含空格分隔的两个正整数 t_i 和 c_i，分别表示第 i 块区域开垦耗时和将耗时缩短 1 天所需资源数量。

【输出格式】

输出到标准输出。

输出一个整数，表示开垦 n 块区域的最少耗时。

【样例 1 输入】

```
4 9 2
6 1
5 1
6 2
7 1
```

【样例 1 输出】

```
5
```

【样例 1 解释】

如下表所示，投入 5 单位资源即可将总耗时缩短至 5 天。此时顿顿手中还剩余 4 单位资源，但无论如何安排，也无法使总耗时进一步缩短。

i	基础耗时 t_i	缩减 1 天所需资源 c_i	投入资源数量	实际耗时
1	6	1	1	5
2	5	1	0	5
3	6	2	2	5
4	7	1	2	5

【样例 2 输入】

```
4 30 2
6 1
5 1
6 2
7 1
```

【样例 2 输出】

```
2
```

【样例 2 解释】

投入 20 单位资源，恰好可将所有区域开垦耗时均缩短为 $k = 2$ 天；受限于 k，剩余的 10 单位资源无法使耗时进一步缩短。

【子任务】

70% 的测试数据满足：$0 < n, t_i, c_i \leq 100$ 且 $0 < m \leq 10^6$；

全部的测试数据满足：$0 < n, t_i, c_i \leq 10^5$ 且 $0 < m \leq 10^9$。

LDAP（ldap）

【题目背景】

西西艾弗岛运营公司是一家负责维护和运营岛上基础设施的大型企业，拥有数千名员工。公司内有很多 IT 系统。为了能够实现这些 IT 系统的统一认证登录，公司 IT 部门决定引入一套 LDAP 系统来管理公司内的用户信息。轻型目录访问协议（Lightweight Directory Access Protocol，LDAP）是一种用于访问和维护目录服务的应用层协议，基于它的数据库可以用树形结构来组织和存储数据。每一笔数据，都包含了一个唯一的标识符（DN，Distinguished Name），以及一系列的属性（Attribute）。

不同的 IT 系统，允许访问的用户是不相同的。每个信息系统都有一个表达式，用来描述允许访问的用户。这个表达式可以按照某一个属性的值作为条件来匹配用户，也可以用多个条件的逻辑组合来匹配用户。小 C 被安排来实现这样一个算法，给定一个 IT 系统的匹配表达式，找到所有与之匹配的用户的 DN。

【题目描述】

为了简化该问题，我们约定，每个用户的 DN 是一个正整数，且不会重复。有若干种用户的属性，用正整数编号。每个用户可以具有这些属性中的若干个，且每个属性只能有一个值。每个属性的值也是一个正整数。例如，假定有两个用户：用户 1 和用户 2，他们的 DN 分别是 1 和 2。一共有 3 种属性。用户 1 具有属性 1 和属性 2，且属性 1 的值为 2，属性 2 的值为 3；但不具有属性 3。用户 2 具有属性 2 和属性 3，且属性 2 的值为 3，属性 3 的值为 1；但不具有属性 1。如下表所示：

DN	属性 1	属性 2	属性 3
1	2	3	N/A
2	N/A	3	1

一个匹配表达式可以是一个属性的值，也可以是多个匹配表达式的逻辑组合。只匹配一个属性的值的表达式称为原子表达式，原子表达式的形式为 <属性编号><操作符><属性值>。其中操作符有两种：断言与反断言。断言操作符为 :，表示匹配具有该属性且值与之相等的用户；反断言操作符为 ~，表示匹配具有该属性且值与之不等的用户。例如，表达式 1:2 可以与上述用户 1 相匹配，但不能与用户 2 相匹配；而表达式 3~1 则不能与任何一个用户相匹配。

表达式可以进行逻辑组合，其语法是：<操作符>(表达式 1)(表达式 2)。其中操作符有两种：与（&）和或（|）。如果操作符为与，则当且仅当两个表达式都与某一用户相匹配时，该表达式与该用户相匹配；如果操作符为或，则当且仅当两个表达式中至少有一个与某一用户相匹配时，该表达式与该用户相匹配。例如，表达式 &(1:2)(2:3) 可

以与用户 1 相匹配，但不能与用户 2 相匹配；而表达式 `|(1:2)(3:1)` 则可以与两个用户都相匹配。

形式化地，上述语法用 BNF 范式表示如下：

```
NON_ZERO_DIGIT =  "1" / "2" / "3" / "4" /
                  "5" / "6" / "7" / "8" / "9"
DIGIT          =  "0" / NON_ZERO_DIGIT
NUMBER         =  NON_ZERO_DIGIT / (NON_ZERO_DIGIT DIGIT*)
ATTRIBUTE      =  NUMBER
VALUE          =  NUMBER
OPERATOR       =  ":" / "~"
BASE_EXPR      =  ATTRIBUTE OPERATOR VALUE
LOGIC          =  "&" / "|"
EXPR           =  BASE_EXPR / (LOGIC "(" EXPR ")" "(" EXPR ")")

EASY_EXPR      =  BASE_EXPR /
                  (LOGIC "(" BASE_EXPR ")" "(" BASE_EXPR ")")
```

【输入格式】

从标准输入读入数据。

输入的第一行包含一个正整数 n，表示用户的数目。

接下来 n 行，每行包含空格分隔的若干个正整数，第一个正整数表示该用户的 DN，第二个正整数表示该用户具有的属性个数，此后的每两个正整数表示该用户具有的一个属性及其值。这些属性按照属性编号从小到大的顺序给出。

接下来一行包含一个正整数 m，表示匹配表达式的数目。

接下来 m 行，每行包含一个匹配表达式。

【输出格式】

输出到标准输出。

输出 m 行，每行包含零个或多个正整数，用空格分隔，表示与对应的匹配表达式相匹配的用户的 DN，由小到大排序。

【样例 1 输入】

```
2
1 2 1 2 2 3
2 2 2 3 3 1
```

```
4
1:2
3~1
&(1:2)(2:3)
|(1:2)(3:1)
```

【样例 1 输出】

```
1

1
1 2
```

【样例 1 解释】

本组输入是题目描述中的例子。

【子任务】

对于 20% 的输入，有 $1 \leq n \leq 100$，$1 \leq m \leq 10$，每个用户的属性个数不超过 10，全部属性编号不超过 100，且表达式是原子表达式，即符合 BNF 语法 BASE_EXPR。

对于 40% 的输入，有 $1 \leq m \leq 100$，每个用户的属性个数不超过 10，全部属性编号不超过 100，且表达式中至多含有两个原子表达式的逻辑组合，即符合 BNF 语法 EASY_EXPR。

对于 70% 的输入，有全部属性编号不超过 500。

对于全部输入，有 $1 \leq n \leq 2500$，$1 \leq m \leq 500$，每个用户的属性个数不超过 500，全部属性编号和属性值均不超过 10^9，每个表达式语句都符合题设语法，且语句字符长度不超过 2000。

星际网络 II（network）

【题目描述】

随着星际网络的进一步建设和规模的增大，一个新的问题出现在网络工程师面前——地址空间不够用了！原来，星际网络采用了传统的 IPv6 协议，虽然有 2^{128} 级别的可用地址数量，但面对广袤无垠的宇宙和爆炸式增长的网络用户数，如此庞大的地址空间也面临了用尽的那一天。

新的通信协议的研发工作交给了著名的网络科技圣地——西西艾弗星。最终，经过 2333 年的不懈努力，西西艾弗星的工程师们设计出了一种新的协议——“西西艾弗 IP 协议”，又称 IPxxaf。

在 IPxxaf 协议中，一个地址由 n 位二进制位组成，其中 n 是 16 的倍数。日常表示一个地址时，采用类似 IPv6 协议的十六进制表示法，每 4 位用 `:` 隔开。如 $n = 32$ 时，地址为 `2a00:0001`，即表示一个二进制为 `0010 1010 0000 0000 0000 0000 0000 0001` 的地址。注意不会出现 IPv6 中省略每组的前导 `0` 或用 `::` 省略一段 `0` 的情况。

为方便起见，记 $num(s)$ 为地址 `s` 按高位在前、低位在后组成的 n 位二进制数，称一段“连续的地址“为 $num(s)$ 成一段连续区间的一系列地址。

西西艾弗星的网络管理员负责地址的分配与管理。最开始，整个地址空间都是未分配的。用户可以随时向管理员申请一些地址：

`1 id l r`：表示用户 id 申请地址在 $l \sim r$ 范围内（包含 l 和 r，下同）的一段连续地址块。

在地址申请操作中，管理员需要先检查地址是否可用。如果用户申请的地址全部未被分配，则检查通过；若地址中存在已经分配给其他用户的地址，则检查失败。

但有一种特殊情况：申请的地址中没有已经分配给其他用户的地址，但含有一些先前已分配给该用户本人的地址。此时可以认为检查通过，但若申请的地址先前已全部分配给该用户则检查失败。

如果上述检查通过，则管理员向用户返回 `YES`，并将申请的地址分配给该用户；若不通过，则向用户返回 `NO`，同时不改变现有的地址分配。

网络管理员要定期检查地址的分配情况，具体而言有如下两种操作：

`2 s`：检查地址 s 被分配给了哪个用户。若未被分配，则结果为 0。

`3 l r`：检查 $l \sim r$ 范围内的所有地址是否完整地分配给了某个用户。若是，回答该用户的编号；若否，回答 0。

在整个网络的运行过程中，共出现了 q 次申请地址和检查地址分配的操作。作为西西艾弗星的一名重要的网络技术顾问，你要帮网络管理员依次处理每个操作，并回答相应的结果。

【输入格式】

从标准输入读入数据。

第一行，2 个正整数 n, q。

接下来 q 行，每行一个操作，格式如上所述，其中的 id 为正整数，l, r, s 均为 IPxxaf 地址串，其中十六进制均用数字和小写字母表示。

【输出格式】

输出到标准输出。

输出 q 行，每行一个非负整数或字符串，表示此次操作的结果。

其中，对于操作 1 ，输出 YES 或 NO；对于操作 $2, 3$，输出一个非负整数。

【样例 1 输入】

```
32 12
1 1 0001:8000 0001:ffff
2 0001:a000
3 0001:c000 0001:ffff
1 2 0000:0000 000f:ffff
2 0000:1000
1 1 0001:8000 0001:8fff
1 2 0000:0000 0000:ffff
2 0000:1000
1 1 0002:8000 0002:ffff
3 0001:8000 0002:ffff
1 1 0001:c000 0003:ffff
3 0001:8000 0002:ffff
```

【样例 1 输出】

```
YES
1
1
NO
0
NO
YES
2
YES
0
```

```
YES
1
```

【样例 1 解释】

第 4 个操作时，由于用户 2 申请的部分地址已被分配给用户 1，因此申请不通过；

第 6 个操作时，由于用户 1 申请的全部地址已被分配给用户 1，因此申请不通过；

第 11 个操作时，用户 1 申请的部分地址已被分配给用户 1，其余地址尚未被分配，申请通过；

【数据范围】

对于所有数据，$n \leq 512$, $q \leq 5 \times 10^4$，n 为 16 的倍数，$id \leq q$，对于操作 $1, 3$ 保证 $num(l) \leq num(r)$。

测试点编号	$n \leq$	$q \leq$	特殊性质
$1 \sim 4$	16	200	无
$5 \sim 6$	64		
$7 \sim 9$	512		
$10 \sim 11$	16	20000	
$12 \sim 13$	64	50000	
$14 \sim 16$	512		所有操作 1 的 id 互不相同
$17 \sim 20$			无

施肥（fertilize）

【题目描述】

春天到了，西西艾弗岛上的 n 块田地需要施肥了。n 块田地编号为 $1, 2, \cdots, n$，按照编号从小到大的顺序排成一列。

为了给田地施肥，顿顿准备了 m 辆施肥车。但是由于土地的松软程度不同，施肥车的质量不一，不一定每一辆施肥车都能给每一块田地施肥。其中，第 i 辆施肥车**只能恰好**从第 l_i 块田地开到第 r_i 块田地，并给编号在 l_i 与 r_i 之间的田地（包含 l_i 和 r_i）都施一遍肥。其中 $1 \le l_i < r_i \le n$。

顿顿希望制定一个施肥的计划。首先，他将选定二元组 $(L, R)(1 \le L < R \le n)$，并选择**只**给编号在 L, R 之间（包含 L, R）的田地施肥。接着，他会从使用这 m 辆施肥车中的一部分（或全部）对田地施肥。他想要保证：**编号在 L 和 R 之内的田地至少被某一辆施肥车施了一次肥，且编号范围外的田地都没有被施过肥。**

现在，他想知道，他能够选择多少种不同的二元组 (L, R) 作为施肥范围，使得可以选出一部分（或全部）施肥车，完成他的目标。

【输入格式】

从标准输入读入数据。

第一行输入两个正整数 n, m，表示田地的块数和施肥车的辆数。数据保证 $2 \le n \le 2 \cdot 10^5, 1 \le m \le 2 \cdot 10^5$。

接下来 m 行，第 i 行输入两个正整数 l_i, r_i，表示第 i 辆施肥车的施肥范围从第 l_i 块田地到第 r_i 块田地。数据保证 $1 \le l_i < r_i \le n$。

【输出格式】

输出到标准输出。

输出一个正整数，表示顿顿能够选择多少种不同的二元组 (L, R) 作为施肥范围，使得他可以选出一部分（或全部）施肥车，完成他的目标。

【样例 1 输入】

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

【样例 1 输出】

```
6
```

【样例 1 解释】

在这组样例中，顿顿可以选择 6 种不同的二元组 (L, R)。
第一种：选择 $(L, R) = (1, 2)$，并只选取第 1 个施肥车施肥。
第二种：选择 $(L, R) = (3, 4)$，并只选取第 2 个施肥车施肥。
第三种：选择 $(L, R) = (2, 3)$，并只选取第 3 个施肥车施肥。
第四种：选择 $(L, R) = (1, 4)$，并选取第 1 个和第 2 个施肥车施肥。
第五种：选择 $(L, R) = (1, 3)$，并选取第 1 个和第 3 个施肥车施肥。
第六种：选择 $(L, R) = (2, 4)$，并选取第 2 个和第 3 个施肥车施肥。

【样例 2】

见题目目录下的 ***2.in*** 与 ***2.ans***。
这个样例满足 $n, m \leq 18$。

【样例 3】

见题目目录下的 ***3.in*** 与 ***3.ans***。
这个样例满足 $n, m \leq 50$。

【样例 4】

见题目目录下的 ***4.in*** 与 ***4.ans***。
这个样例满足 $n, m \leq 400$。

【样例 5】

见题目目录下的 ***5.in*** 与 ***5.ans***。
这个样例满足 $n, m \leq 3000$。

【样例 6】

见题目目录下的 ***6.in*** 与 ***6.ans***。
这个样例满足特殊性质 A。

【样例 7】

见题目目录下的 ***7.in*** 与 ***7.ans***。
这个样例满足 $n, m \leq 200000$。

【子任务】

测试点编号	$n \leq$	$m \leq$	特殊性质
1	18	18	
2	18	18	
3	18	18	
4	50	50	
5	50	50	
6	400	400	
7	400	400	
8	3000	3000	
9	3000	3000	
10	3000	3000	
11	3000	3000	
12	3000	3000	
13	200000	200000	A
14	200000	200000	A
15	200000	200000	A
16	200000	200000	
17	200000	200000	
18	200000	200000	
19	200000	200000	
20	200000	200000	

特殊性质 A：保证任意两个施肥车的施肥范围不存在相互包含的关系，也就是说，对任意 $1 \leq i < j \leq m$，$l_i < l_j, r_i < r_j$ 或 $l_i > l_j, r_i > r_j$。