

# 2024 CCF 非专业级软件能力认证——

## CSP-J/S 2024 第二轮认证

### 提高级 模拟赛 (2024.08.09)

题目	抓拍	杂交向日葵	农庄	简单图论题
题目类型	传统型	传统型	传统型	传统型
目录	photo	hybrid	farm	race
可执行文件名	photo	hybrid	farm	race
输入文件名	photo.in	hybrid.in	farm.in	race.in
输出文件名	photo.out	hybrid.out	farm.out	race.out
每个测试点时限	1.0秒	2.0秒	3.0秒	1.0秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++ 语言	photo.cpp	hybrid.cpp	farm.cpp	race.cpp
----------	-----------	------------	----------	----------

编译选项

对于C++ 语言	-O2 -std=c++14 -static
----------	------------------------

注意事项 (请仔细阅读)

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数main()的返回值类型必须是int，程序正常结束时的返回值必须是0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 全国统一评测时采用的机器配置为：Intel(R)Core(TM)i7-8700K CPU@3.70GHz，内存32GB。  
上述时限以此配置为准。
9. 只提供Linux格式附加样例文件。
10. 评测在当前最新公布的NOI Linux 下进行，各语言的编译器版本以此为准。

# 抓拍 (photo)

## 【题目描述】

学校里有  $n$  名同学在操场上散步。在时间  $t = 0$  秒时，每名同学的坐标可以用  $(x_i, y_i)$  表示，且每个同学的速度为每秒钟一单位长度。

小 A 发现同学们在散步的时候只会向东南西北其中之一的方向行进，且方向固定不变，假设某一位同学正位于  $(x, y)$ ，则下一秒

- 如果向东走，到达  $(x + 1, y)$
- 如果向西走，到达  $(x - 1, y)$
- 如果向南走，到达  $(x, y - 1)$
- 如果向北走，到达  $(x, y + 1)$

作为摄影爱好者，小 A 想在一个**非负整数**秒时抓拍一张照片，一张照片的范围可以用一个四边与坐标轴平行的矩形表示，如果一个同学当前的坐标在矩形内部或边界上，则认为抓拍到了这个同学。

如果同学们一直按照上述规则走下去（忽略碰撞，即允许某时刻多个同学在同一个点），操场的大小也为无限大，那么一定存在某个时刻，包含所有同学的照片取得最小的周长。请帮助小 A 求出这个最小周长的值。

## 【输入格式】

从文件 `photo.in` 中读入数据。

输入的第一行包含一个正整数  $n$ ，表示同学人数。

接下来  $n$  行，每行包含两个正整数  $x_i, y_i$  和一个字符  $d_i$ ，代表每个同学在 0 时刻的位置和移动方向，其中 `E, W, S, N` 分别代表东西南北

## 【输出格式】

输出到文件 `photo.out` 中。

输出一行包含一个正整数，代表包含所有同学的照片的最小周长。

## 【样例1输入】

```
1 5
2 -4 1 E
3 2 -2 N
4 4 -1 W
5 -3 0 E
6 1 0 N
```

## 【样例1输出】

```
1 14
```

### 【样例1解释】

当  $t = 2$  时，每个同学的坐标分别为  $(-2, 1), (2, 0), (2, -1), (-1, 0), (1, 2)$ ，此时能够抓拍到所有同学的周长最小的照片是一个左上角为  $(-2, 2)$ ，右下角为  $(2, -1)$  的矩形，此时周长为  $4 + 3 + 4 + 3 = 14$ 。可以证明这是最小的周长。

### 【样例2】

见选手目录下的 `photo/photo2.in` 与 `photo/photo2.ans`。

### 【数据范围】

对于所有数据，有  $1 \leq n \leq 2 \times 10^5, -10^9 \leq x_i, y_i \leq 10^9$ 。

测试点	$n \leq$	$ x_i ,  y_i  \leq$
1 ~ 3	500	500
4 ~ 10	$2 \times 10^5$	$10^9$

# 杂交向日葵 (hybrid)

## 【题目描述】

小 B 玩到了植物大战僵尸杂交版的新版本，这个版本更新了两种新的向日葵。在某一个固定选卡的挑战关中，卡槽中只有这两种向日葵各一张卡牌，同时关卡要求**最后剩余**一定数量的阳光才能通关。

已知关卡初始提供  $s$  单位的阳光，这两种向日葵的需要花费  $c_1, c_2$  单位的阳光才能种植，且会在种植后每 25 秒分别产出  $s_1, s_2$  单位的阳光。这两种向日葵的冷却时间恰好也是 25 秒，方便起见，我们将每 25 秒称作一轮。假设草坪尺寸无限，不考虑向日葵被吃掉的情况，由于是黑夜关卡，也没有阳光的额外来源，种植只能靠初始阳光以及向日葵产出的阳光。

小 B 是游戏高手，可以将关卡控制在恰好  $n$  轮后结束，阳光视作在每一轮结束的瞬间获取，同时向日葵的种植可以看作在每一轮开始的瞬间完成。请帮助小 B 找到一种策略，使得关卡结束后剩余的阳光最多，求出这个值。

## 【输入格式】

从文件 `hybrid.in` 中读入数据。

第一行包含两个正整数  $s, n$ ，代表初始阳光数量和游戏轮数。

第二行包含两个正整数  $c_1, s_1$ ，代表第一种向日葵的花费和每轮的产出。

第三行包含两个正整数  $c_2, s_2$ ，代表第二种向日葵的花费和每轮的产出。

## 【输出格式】

输出到文件 `hybrid.out` 中。

输出一行包含一个正整数，代表最后剩余阳光的最大值。

## 【样例1输入】

```
1 | 75 3
2 | 50 25
3 | 25 15
```

## 【样例1输出】

```
1 | 125
```

## 【样例1解释】

第一轮开始的时候种两种向日葵各一株，剩余阳光为 0，该轮结束后生产  $15 + 25 = 40$  阳光，共计 40 阳光。

第二轮开始时再种一株 25 阳光的向日葵，剩余阳光为 15，该轮结束后生产  $15 + 15 + 25 = 55$  阳光，共计 70 阳光。

最后一轮不种向日葵，该轮结束后生产  $15 + 15 + 25 = 55$  阳光，共计 125 阳光。

## 【样例2】

见选手目录下的 `hybrid/hybrid2.in` 与 `hybrid/hybrid2.ans` 。

## 【数据范围】

对于所有数据, 有  $1 \leq s, n \leq 2 \times 10^7, 1 \leq c_i, s_i \leq 1023$ 。

不一定保证  $s \geq \min\{c_1, c_2\}$ 。

测试点	$n \leq$
1 ~ 8	15
9 ~ 20	$2 \times 10^7$

# 农庄 (farm)

## 【题目描述】

小 C 正在玩一款经营农庄的游戏，游戏的背景建立在末世，玩家需要以自己的基地为起点，通过开发地下被掩藏的农田以生存下去。

小 C 为了节省额外的通道钱，每次开发新农田时只会以自己已有的一个农田（或基地）为起点向深层挖掘一条通道，这也意味着所有的农田形成了一个以基地为根的树，树上深度越大的节点代表的农田深度也越大。不同农田种植了不同种类及规模的作物，每一片农田只种植一种作物，具有作物营养价值  $p_i$  与作物总量  $w_i$  两个属性，**作物不可再生**。

根据游戏设定，对于同一条通道连接的农田，**深层农田中作物的营养价值总比浅层农田价值低**。某一层的粮食产出可以由以下公式计算：

$$\text{粮食产出} = \sum_i \text{农作物的营养价值 } p_i \times \text{农作物重量 } w_i$$

此外，极少数变异蝗虫可能钻入地下获取食物，由于小 C 隔离措施做的较好，每次获取食物时会只清空一块农田的所有农作物。

小 C 初始只有基地所在位置有一块农田，编号为 0。在每天的日常任务结束以后，小 C 会从编号为  $k$  的农田按最短路径回到基地，在经过每个农田时会选择是否带走其中一些农作物。由于背包空间有限，小 C 希望每次背包中的农作物是目前能够选取的农作物中营养价值最大的一批，请帮助小 C 求出每次回基地时能够带回的最大粮食产出。

## 【输入格式】

从文件 `farm.in` 中读入数据，每个测试点包含多组测试数据。

输入的第一行包含一个正整数  $T$ ，表示数据组数。

每组数据的第一行包含三个正整数  $n, p_0, w_0$ ，代表活动数量，基地所在位置农田里作物的营养价值与重量。

接下来  $n$  行每行首先输入一个  $1 \sim 3$  之间的整数  $opt$ ，代表活动的种类。

- 若  $opt = 1$ ，则代表开发深层农田，接下来给出四个整数  $u, v, p_u, w_u$ ，代表新发现了农田  $u$ ， $u$  和此前以被发现的农田  $v$  有一个通道，以及其中农作物的价值和重量分别为  $p_u, w_u$ 。保证农田  $v$  已经被发现，且  $p_u < p_v$ 。
- 若  $opt = 2$ ，则代表回家，接下来给出两个整数  $k, s$ ，代表从  $k$  号农田出发沿最短路回到基地，且小 C 的背包容量为  $s$ 。
- 若  $opt = 3$ ，则代表蝗虫入侵，接下来给出一个整数  $l$ ，代表  $l$  号农田内的作物被清空。

## 【输出格式】

输出到文件 `farm.out` 中。

对于每个  $opt = 2$  的操作，输出一行两个正整数  $x, y$ ，代表背包中装入的农作物总重量与背包中的粮食产出。

### 【样例1输入】

```
1 1
2 5 6 4
3 2 0 3
4 1 2 0 3 2
5 2 2 2
6 1 4 0 1 3
7 2 4 10
```

### 【样例1输出】

```
1 3 18
2 2 9
3 3 3
```

### 【样例1解释】

第一次活动为从 0 号节点返回基地，能够获取农作物的节点为 0 号点，作物价值为 6，剩余重量为 4。因为背包总量为 3，故全部装这一种作物，农田 0 剩余作物重量为 1。

第二次活动发现了 2 号农田与 0 号农田连接，作物价值为 3，剩余重量为 2。

第三次活动从 2 号节点返回基地，能够获取农作物的节点为 0, 2 号点，作物价值分别为 6, 3，剩余重量分别为 1, 2。由于小 C 想获取最大的产出，故优先选择获取 0 号点的农作物，最终从 0 号点获取了一个单位的农作物，从 2 号点获取了一个单位的农作物，总产出为 9。此时 0 号点不存在农作物，2 号点剩余一个单位的农作物。

第二次活动发现了 4 号农田与 0 号农田连接，作物价值为 1，剩余重量为 3。

第三次活动从 4 号节点返回基地，能够获取农作物的节点为 4 号点，作物价值为 1，剩余重量为 3。小 C 选择全部带走，总产出为 3，背包没有装满，其中农作物的重量也为 3。

### 【样例2】

见选手目录下的 `farm/farm2.in` 与 `farm/farm2.ans`。

### 【数据范围】

对于所有数据，有  
 $1 \leq T \leq 10, 1 \leq n \leq 2 \times 10^5, 1 \leq u \leq 2 \times 10^5, 0 \leq v, l, k \leq 2 \times 10^5, 0 \leq p_i, w_i, s \leq 10^6$ 。

$\sum n$  的值未作限制。

测试点	$n \leq$	特殊性质1	特殊性质2
1 ~ 4	$5 \times 10^4$	保证数据随机生成	2, 3 操作在 1 操作之后出现
5 ~ 8	$5 \times 10^4$	保证 $u=v+1$	2, 3 操作在 1 操作之后出现
9 ~ 11	$2 \times 10^5$	保证数据随机生成	无
12 ~ 14	$2 \times 10^5$	保证 $u=v+1$	无

测试点	$n \leq$	特殊性质1	特殊性质2
15 ~ 20	$2 \times 10^5$	无	无



# 简单图论题 (race)

## 【题目描述】

小 D 生活的国家由  $n$  个城市组成，它们之间由  $m$  条道路连接，每条道路的长度均为 1。小 D 有一个旅行计划，他打算从位于 1 号城市的家出发，访问一个与 1 号城市相邻的节点  $v$ ，再从  $v$  出发到另一个城市  $u$ 。

不过在从城市  $v$  出发之前，小 D 需要提前做好规划。他认为每个城市  $u$  带给他的期待值为  $a_u$ ，但是长途旅行过程会降低他的期待值。具体而言，每走过一条道路，小 D 对城市  $u$  的期待值就会降低  $b_u$ 。最终小 D 到达城市  $u$  会获取的愉悦值为  $a_u - b_u \times d(u, v)$ ，其中  $d(u, v)$  代表从  $u$  到  $v$  的最短距离。

小 D 打算对与城市 1 相邻的每个城市  $v$  都做一个规划，请帮助小 D 求出从  $v$  出发可以获取的最大愉悦值。

简言之，给定一张  $n$  个点  $m$  条边的无向连通图，每个点有属性  $a_u, b_u$ ，请对与 1 相邻的每个节点  $v$  求：

$$\max_{u \neq v} \{a_u - b_u \times d(u, v)\}$$

## 【输入格式】

从文件 `race.in` 中读入数据。

输入的第一行包含两个正整数  $n, m$ ，代表无向图的节点数与边数。

接下来  $n$  行每行包含两个正整数  $a_u, b_u$ ，代表每个节点的属性。

接下来  $m$  行每行包含两个正整数  $u, v$ ，代表图中的一条无向边。

## 【输出格式】

输出到文件 `race.out` 中。

输出一行，对于所有与 1 号点相邻的节点，输出一个正整数代表题目要求的值，按节点编号从小到大的顺序输出。

## 【样例1输入】

```
1 5 4
2 0 0
3 1 1
4 1 1
5 5 1
6 100 40
7 4 1
8 1 2
9 1 3
10 4 5
```

### 【样例1输出】

1	3
2	3
3	60

### 【样例1解释】

与 1 号点相邻的节点为 2, 3, 4, 对于  $v = 2, 3$  时, 选择  $u = 4$ , 愉悦值为  $5 - 1 \times 2 = 3$ , 对于  $v = 4$  时, 选择  $u = 5$ , 愉悦值为  $100 - 40 \times 1 = 60$ 。

### 【样例2】

见选手目录下的 `type/type2.in` 与 `type/type2.ans`。

### 【数据范围】

对于所有数据, 有  $1 \leq n, m \leq 3 \times 10^5$ ,  $1 \leq u, v \leq n$ ,  $0 \leq a_i, b_i \leq 10^9$ 。

保证图连通且无重边自环。

测试点	特殊性质
1 ~ 4	保证所有点都与 1 号节点相邻
5 ~ 8	$m = n - 1$
9 ~ 14	保证数据随机生成
15 ~ 20	无