

Day 2

题目名称	收益	最短路	未来程序
可执行文件名	value	distance	crack
输入文件名	value.in	distance.in	crack.in
输出文件名	value.out	distance.out	crack.out
每个测试点时限	2 秒	1 秒	1 秒
内存限制	256MB	256MB	256MB
是否有部分分	否	否	否
题目类型	传统型	传统型	提交答案型
是否有样例文件	否	否	否
是否有附加文件	否	否	否

提交源程序须加后缀

对于 C++ 语言	value.cpp	distance.cpp	crack.cpp
-----------	-----------	--------------	-----------

编译开关

对于 C++ 语言	-O2 -lm	-O2 -lm	-O2 -lm
-----------	---------	---------	---------

收益(Value)

【问题描述】

给两个 n 维向量 $a, b, \forall 0 \leq i < n; a_i \geq b_i$ 。

你需要设定两个参数 $x, y, x \geq y$ ，对于每个 i ，计算收益：

- 若 $x \leq a_i$ ，获得 x 的收益
- 否则，若 $y \leq b_i$ ，获得 y 的收益
- 否则，获得0的收益

最大化总收益。

【输入格式】

输入文件为 *value.in*。

第一行包含1个正整数 n 。

接下来 n 行每行包含2个整数 a_i, b_i 。

【输出格式与部分分】

输出文件为 *value.out*。

共一行包含1个整数，表示总收益的最大值。

【样例1输入】

```
5
80 20
60 50
40 40
15 10
70 30
```

【样例1输出】

```
220
```

【子任务】

对于100%的数据， $0 \leq b_i \leq a_i \leq 10^9$

编号	分值	n	特殊性质
1	16	≤ 3000	
2	18	≤ 100000	$a = b$
3	11	≤ 50000	
4	12	≤ 75000	
5	43	≤ 150000	

最短路(Distance)

【问题描述】

给定一个 $n \times m$ 的网格图和包括 $(1,1), (n,m)$ 在内的 k 个标记点, 你只能通过标记点进行转移。

任意两个标记点间的距离定义为2的切比雪夫距离次方, 即 (x_1, y_1) 和 (x_2, y_2) 间的距离为 $2^{\max(|x_1-x_2|, |y_1-y_2|)}$ 。

输出一条从 $(1,1)$ 到 (n,m) 的最短路。

【输入格式】

输入文件为 *distance.in*。

第一行包含3个正整数 n, m, k 。

接下来 k 行每行包含2个正整数 x_i, y_i , 表示一个标记点的坐标, 其中1号点一定是 $(1,1)$, k 号点一定是 (n,m) 。

【输出格式与部分分】

输出文件为 *distance.out*。

第一行包含一个正整数 t 。

第二行包含 t 个正整数 $ans_1, ans_2, \dots, ans_t$, 表示一条最短路。 ans_i 表示最短路中到达的第 i 个标记点的编号, 其中 ans_1 一定是1, ans_t 一定是 k 。

【样例 1 输入】

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

【样例 1 输出】

```
5
```

1 6 2 8 9

【子任务】

所有测试数据的范围和特点如下：

对于 100 % 的数据， $1 \leq x_i \leq n$ ， $1 \leq y_i \leq m$

编号	分值	n, m, k	特殊性质
1	23	≤ 500	
2	36	≤ 10000	每行每列最多一个标记点
3	41	≤ 10000	

未来程序(Crack)

【问题背景】

本题共分为 10 个子任务，下文将有针对每个子任务的具体说明。每个子任务各对应一种不同的基础算法，并且保证采用的任意单个算法对所有合法输入数据具有普适性，即不会针对有限组输入数据采用特例。你的目标便是还原这些算法。

为了发掘这些程序所完成的任务，你可以利用下发的可执行文件 `crack_force` 运行任意自行设计的合法输入数据，具体使用方法将在下文提及。然而由于某些原因，下发的 `crack_force` 只能处理规模较小的输入数据，并且将会花费较多的时间。但你还原的程序必须在规定时间内解决规模较大的问题。

`crack_force` 从 `crack_force.in` 读取输入数据，并将输出数据写入到 `crack_force.ans`。

本题将采用传统题方式进行评测，针对每个子任务均有若干组赛前已经生成的测试点，以检验你还原出的算法的正确性，只有全部通过才能获得该子任务的全部分数。

【输入格式】

输入文件为 `crack.in`。

对于所有数据，输入第一行均为 1 个整数 t ， $1 \leq t \leq 10$ ，表示该组数据所属的子任务编号。

接下来对于每个子任务分别进行描述（描述中出现的变量名均按照小写字母的顺序依次标识）：

➤ 子任务 1\2

- 共一行包含 1 个整数 a 。

➤ 子任务 3

- 第一行包含 2 个整数 a, b 。
- 第二行包含 a 个整数 c_1, c_2, \dots, c_a ， $-10^9 \leq c_i \leq 10^9$ 。
- 接下来 b 行每行包含 2 个整数 d_i, e_i ， $1 \leq d_i \leq e_i \leq a$ 。

➤ 子任务 4

- 第一行包含 2 个整数 a, b 。
- 第二行包含 a 个整数 c_1, c_2, \dots, c_a ， $0 \leq c_i \leq 10^9$ 。

- 接下来 b 行每行包含2个整数 d_i, e_i , $1 \leq d_i \leq e_i \leq a$ 。
- 子任务 5/6
 - 共一行包含1个整数 a 。
 - 接下来 $a - 1$ 行每行包含3个整数 b_i, c_i, d_i , $1 \leq b_i, c_i \leq a$; $0 \leq d_i \leq 10^9$, 数据满足:
 - ◆ $\forall 1 \leq i \leq a - 1, b_i \neq c_i$
 - ◆ $\forall 1 \leq i, j \leq a - 1, (b_i, c_i) \neq (b_j, c_j) \text{ 且 } (b_i, c_i) \neq (c_j, b_j)$
 - ◆ $\forall 1 \leq i, j \leq a$, 存在序列 $\{e_f\}$ 满足:
 - $e_1 = i$
 - $e_f = j$
 - $\forall 1 \leq k < f$, 存在 $1 \leq l \leq a - 1$ 满足 $(e_k, e_{k+1}) = (b_l, c_l)$ 或 $(e_k, e_{k+1}) = (c_l, b_l)$
- 子任务 7/8
 - 共一行包含2个整数 a, b 。
 - 接下来 b 行每行包含3个整数 c_i, d_i, e_i , $1 \leq c_i, d_i \leq a$; $0 \leq e_i \leq 10^9$, 数据满足:
 - ◆ $\forall 1 \leq i \leq b, c_i \neq d_i$
 - ◆ $\forall 1 \leq i, j \leq b, (c_i, d_i) \neq (c_j, d_j) \text{ 且 } (c_i, d_i) \neq (d_j, c_j)$
 - ◆ $\forall 1 \leq i, j \leq a$, 存在序列 $\{f_g\}$ 满足:
 - $f_1 = i$
 - $f_g = j$
 - $\forall 1 \leq k < g$, 存在 $1 \leq l \leq b$ 满足 $(e_k, e_{k+1}) = (c_l, d_l)$ 或 $(e_k, e_{k+1}) = (d_l, c_l)$
- 子任务 9/10
 - 共一行包含一个由小写字母组成的字符串 a 。

【输出格式】

输出文件为 **crack.out**。

➤ 子任务 1/2/5/6/7/8/9/10

■ 共一行包含一个整数 ans , $-10^{18} \leq ans \leq 10^{18}$ 。

➤ 子任务 3/4

■ 共 b 行每行包含一个整数 ans , $-10^{18} \leq ans \leq 10^{18}$ 。

【子任务】

编号	分值	$crack_force$ 数据范围	$crack$ 数据范围
1	6	$2 \leq a \leq 10^5$	$2 \leq a \leq 10^9$
2	10		
3	7	$2 \leq a, b \leq 10^3$	$2 \leq a, b \leq 10^5$
4	11		$2 \leq a, b \leq 10^4$
5	8	$2 \leq a \leq 100$	$2 \leq a \leq 1000$
6	12		
7	9	$2 \leq a \leq 100$	$2 \leq a \leq 1000$
8	13	$2 \leq b \leq 200$	$2 \leq b \leq 2000$
9	10	$2 \leq a \leq 100$	$2 \leq a \leq 1000$
10	14		