

全国青少年信息学奥林匹克竞赛

CCF NOI2021 统一省选

Day1 (B 卷)

时间：2021 年 4 月 10 日 08:30 ~ 13:00

题目名称	数对	卡牌游戏	图函数
题目类型	传统型	传统型	传统型
目录	pair	card	graph
可执行文件名	pair	card	graph
输入文件名	pair.in	card.in	graph.in
输出文件名	pair.out	card.out	graph.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
子任务数目	10	10	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	pair.cpp	card.cpp	graph.cpp
-----------	----------	----------	-----------

编译选项

对于 C++ 语言	-O2
-----------	-----

注意事项与提醒（请选手务必仔细阅读）

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++ 中主函数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照各省的具体要求。
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. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 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。
8. 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其为准。
9. 最终评测时所用的编译命令中不含编译选项之外的任何优化开关。

数对 (pair)

【题目描述】

给定 n 个正整数 a_i , 请你求出有多少个数对 (i, j) 满足 $1 \leq i \leq n, 1 \leq j \leq n, i \neq j$ 且 a_i 是 a_j 的倍数。

【输入格式】

从文件 *pair.in* 中读入数据。

第一行一个整数 n , 表示数字个数。

第二行 n 个整数表示 a_i 。

【输出格式】

输出到文件 *pair.out* 中。

输出一行一个整数表示答案。

【样例 1 输入】

```
1 6
2 16 11 6 1 9 11
```

【样例 1 输出】

```
1 7
```

【样例 1】

见选手目录下的 *pair/pair1.in* 与 *pair/pair1.ans*。

【数据范围】

对于 40% 的数据: $n \leq 1000$ 。

对于 70% 的数据: $1 \leq a_i \leq 5 \times 10^3$ 。

对于 100% 的数据: $2 \leq n \leq 2 \times 10^5, 1 \leq a_i \leq 5 \times 10^5$ 。

卡牌游戏 (card)

【题目描述】

Alice 有 n 张卡牌, 第 i ($1 \leq i \leq n$) 张卡牌的正面有数字 a_i , 背面有数字 b_i , 初始时所有卡牌正面朝上。

现在 Alice 可以将不超过 m 张卡牌翻面, 即由正面朝上改为背面朝上。Alice 的目标是让最终朝上的 n 个数字的极差 (最大值与最小值的差) 尽量小。请你帮 Alice 算一算极差的最小值是多少。

【输入格式】

从文件 `card.in` 中读入数据。

第一行两个正整数 n, m , 代表卡牌张数与至多翻面张数。

第二行 n 个正整数, 第 i 个数字表示 a_i 。

第三行 n 个正整数, 第 i 个数字表示 b_i 。

数据保证卡牌上的 $2n$ 个数字互不相同, 且卡牌按照 a_i 升序给出。

【输出格式】

输出到文件 `card.out` 中。

仅一行一个整数表示答案。

【样例 1 输入】

```
1 6 3
2 8 11 13 14 16 19
3 10 18 2 3 6 7
```

【样例 1 输出】

```
1 8
```

【样例 1 解释】

最优方案之一: 将第 1, 5, 6 张卡牌翻面, 最终朝上的数字依次为 10, 11, 13, 14, 6, 7, 极差为 $14 - 6 = 8$ 。

【样例 2】

见选手目录下的 *card/card2.in* 与 *card/card2.ans*。

【样例 3】

见选手目录下的 *card/card3.in* 与 *card/card3.ans*。

【数据范围】

对于所有测试数据： $3 \leq n \leq 10^6$ ， $1 \leq m < n$ ， $1 \leq a_i, b_i \leq 10^9$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	特殊限制
1 ~ 2	10	无
3 ~ 4	500	
5 ~ 6	5×10^5	$m \leq 1000$
7	1×10^5	无
8	4×10^5	
9	7×10^5	
10	1×10^6	

图函数 (graph)

【题目描述】

对于一张 n 个点 m 条边的有向图 G (顶点从 $1 \sim n$ 编号), 定义函数 $f(u, G)$:

1. 初始化返回值 $cnt = 0$, 图 $G' = G$ 。
2. 从 1 至 n 按顺序枚举顶点 v , 如果当前的图 G' 中, 从 u 到 v 与从 v 到 u 的路径都存在, 则将 $cnt + 1$, 并在图 G' 中删去顶点 v 以及与它相关的边。
3. 第 2 步结束后, 返回值 cnt 即为函数值。

现在给定一张有向图 G , 请你求出 $h(G) = f(1, G) + f(2, G) + \cdots + f(n, G)$ 的值。

更进一步地, 记删除 (按输入顺序给出的) 第 $1 \sim i$ 条边后的图为 G_i ($1 \leq i \leq m$), 请你求出所有 $h(G_i)$ 的值。

【输入格式】

从文件 `graph.in` 中读入数据。

第一行两个整数 n, m 表示图的点数与边数。

接下来 m 行每行两个整数, 第 i 行的两个整数 x_i, y_i 表示一条有向边 $x_i \rightarrow y_i$ 。

数据保证 $x_i \neq y_i$ 且同一条边不会给出多次。

【输出格式】

输出到文件 `graph.out` 中。

输出一行 $m + 1$ 个整数, 其中第一个数表示给出的完整图 G 的 $h(G)$ 值。第 i ($2 \leq i \leq m + 1$) 个整数表示 $h(G_{i-1})$ 。

【样例 1 输入】

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

【样例 1 输出】

```
1 6 5 5 4 4 4 4
```

【样例 1 解释】

对于给出的完整图 G :

1. $f(1, G) = 1$, 过程中删除了顶点 1。
2. $f(2, G) = 1$, 过程中删除了顶点 2。
3. $f(3, G) = 2$, 过程中删除了顶点 2, 3。
4. $f(4, G) = 2$, 过程中删除了顶点 1, 4。

【样例 2】

见选手目录下的 `graph/graph2.in` 与 `graph/graph2.ans`。

【数据范围】

对于所有测试数据: $2 \leq n \leq 1000$, $1 \leq m \leq 2 \times 10^5$, $1 \leq x_i, y_i \leq n$ 。

每个测试点的具体限制见下表:

测试点编号	$n \leq$	$m \leq$
1 ~ 4	10	10
5 ~ 11	100	2000
12 ~ 20	1000	5000
21 ~ 25		2×10^5