必可赛前公益众筹赛

第一试

时间：2024 年 9 月 23 日 08:00 ～ 12:00

题目名称	多线并行	空当接球	列序号括	可爱路径
题目类型	传统型	传统型	传统型	传统型
目录	multi	ball	bracket	path
可执行文件名	multi	ball	bracket	path
输入文件名	multi.in	ball.in	bracket.in	path.in
输出文件名	multi.out	ball.out	bracket.out	path.out
每个测试点时限	1.0 秒	3.0 秒	1.0 秒	1.0 秒
内存限制	1024 MiB	1024 MiB	1024 MiB	1024 MiB
测试点数目	10	10	10	4
测试点是否等分	是	是	是	否

提交源程序文件名

对于 C++ 语言	multi.cpp	ball.cpp	bracket.cpp	path.cpp

编译选项

对于 C++ 语言	-O2 -std=c++14

注意事项

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 若无特殊说明，每道题的代码大小限制为 100KB。
7. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
8. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 scanf 函数）避免出错。
9. 直接复制 PDF 题面中的跨页样例，数据将带有行序号和页眉页脚，建议选手直接使用对应目录下的样例文件进行测试。
10. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。
11. 若发现有原题，请**保持静默状态直至比赛结束**，否则作违反考试纪律处理，**禁赛 3 年**。

多线并行（multi）

【题目背景】

这是一道签到题。

【题目描述】

小 Z 打模拟赛又被小 L 锤爆了，为了不被小 L 天天嘲讽，于是准备找电饭煲学习经验。小 Z 认真观察了电饭煲做题方法学习，总结出了以下规律。

- 打开若干个题目页面并阅读题面。
- 随机选择顺序开始打注释。
- 全部切掉。

小 Z 决定亲身实践一下。具体来说，小 Z 选出了 n 道题，并对每一道题估测了一个难度值 a_i。小 Z 在某一时刻可以选择开始思考一道题，所需要花费时间为 a_i，在这期间不能思考其他的题。

小 Z 还有一个脑容量 m，当小 Z 开始思考一道题时，会**立刻**占据 a_i 的脑容量。但小 Z 很快会发现自己完全没水平解决这些问题，于是在一道题被思考完成后，小 Z 需要花费 1 的时间说服自己放弃这一道题，这时被占据的 a_i 脑容量会释放。小 Z 坚信一道题在思考完成前是不能放弃的，所以一定会思考完所有题目。

小 L 看到之后又来嘲讽小 Z 了，它想让你帮忙求求小 Z 最少花费多少时间才会放弃所有题。(小 L：shaber 小 Z。)

小 Z 在说服自己放弃一道的同时也可以思考其他的题。

【输入格式】

从文件 ***multi.in*** 中读入数据。

本题有多组数据，输入数据第一行一个正整数 T，表示数据组数。

对于每组数据：

第一行包含两个整数 n, m，表示小 Z 思考的题目数量与脑容量。

第二行包含 n 个整数 a_i，表示每一道题目的难度值。

【输出格式】

输出到文件 ***multi.out*** 中。

输出共有 T 行，每行一个整数，表示花费时间的最小值。

【样例 1 输入】

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

【样例 1 输出】

```
9
```

【样例 1 解释】

一种最优的安排方式是：先思考题目一，花费 2 的时间；思考题目二，同时放弃题目一，花费 2 的时间。由于题目二、三无法同时存在脑子里，放弃题目二，花费 1 的时间；思考题目三并放弃，花费 $3+1=4$ 的时间，共计 9。

【样例 2 输入】

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

【样例 2 输出】

```
16
17
12
```

【样例 3】

见选手目录下的 ***multi/multi3.in*** 与 ***multi/multi3.ans***。

【子任务】

对于 100% 的数据：$1 \le n \le 2 \times 10^5, 1 \le \sum n \le 10^6$, $1 \le a_i \le m \le 1e9$。

测试点	$n \leqslant$	$\sum n \leqslant$
$1 \sim 3$	5	10^6
$4 \sim 6$	50	
$7, 8$	2×10^5	2×10^5
$9, 10$		10^6

空当接球（ball）

【题目背景】

这是一道中档题。

小 Z 玩过很多有趣的游戏：例如著名的《移球游戏》，收获无数好评的《喵了个喵》。

小 Z 想要自己发明一个游戏，于是他想出了一个叫做《空当接球》的游戏。

【题目描述】

小 Z 面前有 n 堆球，每堆球的个数不超过 m 个但至少有 1 个，其中初状态第 i 堆球有 a_i 个球。《空当接球》这个游戏玩法如下，小 Z 可以做下列操作任意次:

- 选择两个整数 $i,j(1 \le i,j \le n, i \ne j)$ 满足当前状态下 $a_i \ge a_j$，小 Z 会将第 i 堆球中的 a_j 个球转移到第 j 堆球中（即 $a_i \leftarrow a_i - a_j$，$a_j \leftarrow 2 \times a_j$）。

小 Z 的目标是最后（末状态）满足至多存在两堆球中有球。小 Z 发现这个游戏十分有趣，但是有太多堆球了，小 Z 的脑子转不过来了，于是想让你写一个程序帮助他。为了增加游戏的难度，小 Z 限制你最多进行 8.5×10^5 次操作。

小 Z 相信你是无比聪明的，同时相信你也能体会到《空当接球》这个游戏的乐趣!

小 Z 发现《空当接球》游戏可以证明一定有解，一定存在一个操作方案使得末状态至多存在两堆球中有球。

【输入格式】

从文件 ***ball.in*** 中读入数据。

第一行包含两个整数 n, m。

第二行包含 n 个整数，第 i 个整数表示初状态第 i 堆的球数 a_i。

【输出格式】

输出到文件 ***ball.out*** 中。

本题采用自定义校验器（Special Judge）评测。

输出的第一行应包含一个整数 k，表示你的方案的操作次数，你应保证 $0 \le k \le 8.5 \times 10^5$。

接下来 k 行，每行包含两个整数 x, y，表示该次操作你选择了 $i = x, j = y$。

【样例 1 输入】

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

【样例 1 输出】

```
2
1 2
2 3
```

【样例 1 解释】

第一次操作后，$a_1 = 0, a_2 = 2, a_3 = 2, a_4 = 4$。
第二次操作后，$a_1 = 0, a_2 = 0, a_3 = 4, a_4 = 4$。
由于特殊原因，本题大样例不下发答案文件，所有 `.ans` 文件均为空。

【样例 2】

见选手目录下的 ***ball/ball2.in*** 与 ***ball/ball2.ans***。
该样例约束与子任务 2 一致。

【样例 3】

见选手目录下的 ***ball/ball3.in*** 与 ***ball/ball3.ans***。
该样例约束与子任务 4 一致。

【样例 4】

见选手目录下的 ***ball/ball4.in*** 与 ***ball/ball4.ans***。
该样例约束与子任务 5 一致。

【样例 5】

见选手目录下的 ***ball/ball5.in*** 与 ***ball/ball5.ans***。
该样例约束与子任务 7 一致。

【样例 6】

见选手目录下的 ***ball/ball6.in*** 与 ***ball/ball6.ans***。
该样例约束与子任务 8 一致。

【样例 7】

见选手目录下的 ***ball/ball7.in*** 与 ***ball/ball7.ans***。
该样例约束与子任务 9 一致。

【样例 8】

见选手目录下的 ***ball/ball8.in*** 与 ***ball/ball8.ans***。
该样例约束与子任务 10 一致。

【子任务】

本题采用捆绑测试。
对于 100% 的测试数据，$3 \le n \le 3 \times 10^5$，$1 \le m \le 10^{10}$。

子任务编号	$n \le$	$m \le$	分值
1	3	10^4	10
2	10	5	
3	5×10^4	1	
4	10^2	10^2	
5	500	500	
6	10^4	10^3	
7	3×10^4		
8	5×10^4		
9		10^9	
10	3×10^5	10^{10}	

其中，子任务 $1 \sim 8$ 保证 $\forall 1 \le i \le n$，a_i 在值域 $[1, m]$ 中均匀随机生成。
小 L 的忠告：或许你想出的依靠随机数据的算法在不随机下也同样适用！
小 Z 是仁慈的，就算你无法达成目标他也想给你一些分数，设末状态存在球的堆数为 K，该测试点分数为 S，计算出 $P = \frac{n-\max(K,2)}{n-2}$，如果 $P = 1$，那么你可以得到 S 的分数，否则你可以得到 $\lfloor 100S \times \min(P + 0.05, 0.8) \rfloor$ 的分数。

【提示】

为了方便选手调试,本题会下发 `Special Judge`,见下发文件中的 `ball/checker.cpp`。将下放的 `checker.cpp` 与 `testlib.h` 放在同一文件目录下，编译 `checker.cpp` 得到 `checker.exe`。打开系统的 `cmd`，将 `checker.exe`，`ball?.in`，`ball?.out`，`ball?.ans` 依次拖入 `cmd`，按回车即可得到结果。其中 `ball?.in` 为输入，`ball?.out` 为选手输出，`ball?.ans` 为标准答案 (当然，此题中标准答案并不必要，使用 `Special Judge` 的时候最后拖入一个空文件当作 `ans` 文件即可，下发的 `spj` 并不会用到标准答案文件)。
你可能会得到如下结果:

- `Wrong Answer receive signal 1.`: 这表示你的操作次数非法。
- `Wrong Answer receive signal 2.`: 存在一次操作选择的 i, j 不属于值域 $[1, n]$。
- `Wrong Answer receive signal 3.`: 存在一次操作选择的 i, j 相等。

- `Wrong Answer receive signal 4.`: 存在一次操作选择的 i, j 不满足在当前状态 $a_i \geq a_j$ 。
- `P1，P2% Score!`: 你得到了 P_1 的分数（满分以 10 分计算），其中 $P_1 = \lfloor 10P_2 \rfloor$。
- `AC!`: 你给出的方案正确，得到满分。

注意，本题输出量很大，请使用下发或自己写的读入/输出优化，防止被卡常数！

列序号括（bracket）

【题目背景】

这是一道字符串题，不采用捆绑测试。

【题目描述】

小 Z 是括号序列的狂热爱好者，对特殊的括号序列更是情有独钟。

由于研究过过多的括号序列，小 Z 已经对各种各样的括号序列计数，权值转换，前后缀求和失去兴趣了，它决定来解决最小字典序问题，下面的描述中，我们钦定左括号的字典序小于右括号的字典序。

小 Z 先生成了一个括号序列，它希望通过若干次操作使得剩下的括号序列字典序最小。

我们定义一次操作可以删除位置 i,j 上的字符当且仅当 $i<j$ 且位置 i 为左括号，位置 j 为右括号。

小 Z 当然是轻松解决了，它想让你写一份程序来验证正确性。

【输入格式】

从文件 ***bracket.in*** 中读入数据。

输入仅包含一行，包含一个字符串 s，表示给定的括号序列。

【输出格式】

输出到文件 ***bracket.out*** 中。

输出只有一行，包含一个字符串 s'，表示经过若干次操作后的字典序最小的括号序列。

【样例 1 输入】

```
1 )(()(()()))
```

【样例 1 输出】

```
1 )((()()))
```

【样例 2 输入】

```
1 )((())
```

【样例 2 输出】

```
1 )(
```

【样例 3】

见选手目录下的 ***bracket/bracket3.in*** 与 ***bracket/bracket3.ans***。

【子任务】

对于 100% 的数据，$1 \leq |s| \leq 10^6$。

测试点编号	$\|s\|$
$1, 2$	≤ 20
$3, 4$	≤ 500
$5, 6$	$\leq 5,000$
$7, 8$	$\leq 10^5$
$9, 10$	$\leq 10^6$

可爱路径（path）

【题目背景】

这是一道图论题，采用捆绑测试。

【题目描述】

这原本是一道简单的最短路问题，但是由于种种地域文化，宗教信仰以及政治因素，原来一些或许可以行走的路径不能通行了。

我们定义禁止路径为**连续**的经过一些特定的点的路径。

一条可爱路径指从 1 到 n 的不经过任何禁止路径的路径。

给定了一张 n 个点 m 条边的有向带权图和 k 条禁止路径，现在小 Z 想让你求出最短的可爱路径，若不存在输出 `-1`。

为什么叫做可爱路径呢，因为这道题原来很可爱。

【输入格式】

从文件 ***path.in*** 中读入数据。

第一行三个整数 n, m, k 分别表示点数，边数，以及禁止路径的数量。

接下来 m 行，每行三个整数 u, v, w 表示一条有向边。

接下来 k 行，每行先输入一个 p 表示这条禁止路径的长度，紧接着 p 个数表示该条禁止路径。

【输出格式】

输出到文件 ***path.out*** 中。

输出只有一行，包含一个整数，表示 1 到 n 的最短可爱路径的长度，若不存在输出 `-1`。

【样例 1 输入】

```
7 8 2
1 2 1
2 3 2
3 4 1
4 5 1
4 6 1
5 7 1
6 7 1
```

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

【样例 1 输出】

```
8
```

【样例 2 输入】

```
4 4 2
1 2 1
2 3 1
3 4 1
3 2 1
4 1 2 3 4
6 1 2 3 2 3 4
```

【样例 2 输出】

```
7
```

【样例 3】

见选手目录下的 ***path/path3.in*** 与 ***path/path3.ans***。

【子任务】

本题采用捆绑测试。

对于 100% 的数据，$1 \le n \le 2\times10^5$，$0 \le m,k \le 2\times10^5$，$0 \le w \le 10^9$，$\sum p \le 2\times10^5$，$1 \le p \le 2\times10^5$。

保证无重边（有序点对不重，$u \to v$ 与 $v \to u$ 这两条边可能同时出现）无自环，不保证数据输入中给出的禁止路径一定可以在原图中找到对应路径。

子任务编号	n	m	k	p	分值
1	$\leq 10^2$	$\leq 10^2$	≤ 0	$\leq 2\times 10^5$	20
2			$\leq 2\times 10^5$	≤ 4	
3	$\leq 2\times 10^5$	$\leq 2\times 10^5$	≤ 1	$\leq 2\times 10^5$	30
4			$\leq 2\times 10^5$		