

模拟赛

2023 年 xx 月 xx 日

题目名称	迷宫探险	划分区间	西西弗西	二分图匹配
题目类型	传统型	传统型	传统型	传统型
可执行文件名	maze	segment	ccfc	match
输入文件名	maze.in	segment.in	ccfc.in	match.in
输出文件名	maze.out	segment.out	ccfc.out	match.out
时间限制	1.0 秒	1.0 秒	1.0 秒	4.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	20	10	20	20

提交源程序文件名

对于 C++	maze.cpp	segment.cpp	ccfc.cpp	match.cpp
--------	----------	-------------	----------	-----------

编译选项

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

注意事项（请仔细阅读）

1. 选手提交的源程序请**放置在以题目对应文件名命名的子文件夹下**。
2. 文件名（包括程序名和输入输出文件名）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. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 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。
6. 若无特殊说明，每道题的**代码大小限制为 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。

迷宫探险 (maze)

【题目描述】

小 P 误入了一个迷宫！迷宫是一个 n 个节点 m 条边的无向图，图中有 k 个节点可以离开迷宫。

图中的每条边 (u_i, v_i) 都有一个边权 w_i ，小 P 可以选择花费 w_i 的时间从这条边的某个端点走到另一个端点。他的目标是走到一个可以离开迷宫的节点。

然而事情没有他想象地那么顺利，图上的每个节点 u 都有 d_u 个魔法荆棘，当小 P 位于节点 u 时这些荆棘会随意占据 u 的 d_u 条连边，使得小 P 无法经过这些边。而当小 P 离开后这些荆棘又会缩回去。

小 P 是一个考虑周全的人，现在小 P 位于 1 节点，他想让你计算最坏情况下小 P 需要花费多少时间才能离开迷宫。或者告诉他最坏情况下不可能离开迷宫。

【输入格式】

输入到文件 `maze.in` 中。

第一行输入三个整数表示 n, m, k 。

第二行输入 k 个整数 s_1, s_2, \dots, s_k 表示所有可以离开迷宫的节点的编号，按照升序顺序给出。

第三行输入 n 个整数 d_1, d_2, \dots, d_n 表示每个节点的荆棘数量。

接下来输入 m 行，每行三个整数 u_i, v_i, w_i 分别表示一条边的两端点，以及经过这条边需要花费的时间。

【输出格式】

输出到文件 `maze.out` 中。

输出一行一个整数表示答案。特别地如果无法离开迷宫那么输出 -1。

【样例 1 输入】

```
5 8 3
2 3 5
2 0 1 1 2
1 2 1
2 3 2
3 4 5
```

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

【样例 1 输出】

```
6
```

【样例 2】

见下发文件中的 *maze/ex_maze2.in* 与 *maze/ex_maze2.ans*。

【样例 3】

见下发文件中的 *maze/ex_maze3.in* 与 *maze/ex_maze3.ans*。

【样例 4】

见下发文件中的 *maze/ex_maze4.in* 与 *maze/ex_maze4.ans*。

【测试点约束】

对于 30% 的数据, 满足 $n, m \leq 15$ 。

对于 50% 的数据, 满足 $n, m \leq 2000$ 。

对于另外 15% 的数据, 满足 $d_u = 0$ 。

对于另外 15% 的数据, 满足 $d_u \leq 1$ 。

对于全部数据, 满足 $1 \leq n, m \leq 2 \times 10^5$, $1 \leq k \leq n$, $1 \leq s_1 < s_2 < \dots < s_k \leq n$, $0 \leq d_i \leq m$, $1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^4$ 。保证给出的图不存在重边或自环。

划分区间 (segment)

【题目描述】

小 P 有一个长度为 n 的序列 a_1, a_2, \dots, a_n ，现在他想将整个序列划分为若干个区间，满足每个区间的长度要么为 1 要么为 2。

定义一个区间的权值为其包含元素的 a_i 的和，现在小 P 想要最小化所有区间权值的极差，请你帮他求出这个值。

一个集合的数的极差定义为其中的最大值减去最小值。

【输入格式】

从文件 *segment.in* 中读入数据。

第一行输入一个整数表示 n 。

第二行输入 n 个整数表示序列 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 *segment.out* 中。

输出一行一个整数表示答案。

【样例 1 输入】

```
5
2 2 3 5 2
```

【样例 1 输出】

```
3
```

【样例 2】

见下发文件中的 *segment/ex_segment2.in* 与 *segment/ex_segment2.ans*。

【样例 3】

见下发文件中的 *segment/ex_segment3.in* 与 *segment/ex_segment3.ans*。

【样例 4】

见下发文件中的 *segment/ex_segment4.in* 与 *segment/ex_segment4.ans*。

【测试点约束】

对于 20% 的数据，满足 $n \leq 18$ 。

对于 40% 的数据，满足 $n \leq 300$ 。

对于 60% 的数据，满足 $n \leq 2000$ 。

对于另外 20% 的数据，满足 $a_i \leq 10$ 。

对于全部数据，满足 $1 \leq n \leq 2 \times 10^5$ ， $1 \leq a_i \leq 10^9$ 。

西西弗西 (ccfc)

【题目描述】

小 P 认为西西弗西是一个非常好听的名字，因为它是 AABA 的格式。

具体地，小 P 可以把所有名字都抽象成仅由小写字母构成的字符串，对于一个名字 S 来说，如果存在两个非空字符串 A, B 使得这两个字符串按照 AABA 的顺序拼接起来后就是字符串 S ，那么小 P 就认为名字 S 是好听的。

仅仅判断是否好听还是不够的，小 P 现在有一个字符串 s ，令 s_i 表示 s 以 i 开头的后缀，现在小 P 想要知道对于 $i = 1, 2, \dots, n$ ，有多少种选择 A, B 的方式使得 $s_i = AABA$ 。

【输入格式】

从文件 *ccfc.in* 中读入数据。

输入一行一个仅有小写英文字母构成的字符串 s 。

【输出格式】

输出到文件 *ccfc.out* 中。

输出一行 $|s|$ 个整数，第 i 个表示 s_i 的答案。

【样例 1 输入】

```
ouvouvouvou
```

【样例 1 输出】

```
0 0 1 0 0 0 0 0 0 0 0 0 0
```

【样例 2】

见下发文件中的 *ccfc/ex_ccfc2.in* 与 *ccfc/ex_ccfc2.ans*。

【样例 3】

见下发文件中的 *ccfc/ex_ccfc3.in* 与 *ccfc/ex_ccfc3.ans*。

【测试点约束】

对于 15% 的数据，满足 $|s| \leq 50$ 。

对于 40% 的数据，满足 $|s| \leq 300$ 。

对于 55% 的数据，满足 $|s| \leq 2000$ 。

对于 70% 的数据，满足 $|s| \leq 10000$ 。

对于全部数据，满足 $1 \leq |s| \leq 10^5$ ， s 仅包含小写英文字母。

二分图匹配 (match)

【题目背景】

这道题的题面本来是和二分图匹配相关的，由于太抽象就改了，于是现在的题面和二分图就没有关系啦。

【题目描述】

有 n 个二进制数，长度不超过 m ，定义为 a_1, a_2, \dots, a_n 。

你需要构造出一组 a 满足：

- $\forall 1 \leq i \leq n, a_i \geq b_i$
- $\forall 1 \leq i < j \leq n, a_i \text{ AND } a_j = 0$

你需要最小化 $s = a_1 \text{ OR } a_2 \text{ OR } \dots \text{ OR } a_n$ ，输出 s 。

【输入格式】

从文件 **match.in** 中读入数据。

第一行输入两个整数表示 n, m 。

接下来 n 行，第 i 行输入若干个整数 k, p_1, p_2, \dots, p_k 用来描述二进制 b_i ，表示 $b_{i,p_1}, b_{i,p_2}, \dots$ 均为 1，其余均为 0。

【输出格式】

输出到文件 **match.out** 中。

输出一行一个长度为 m 的 01 字符串表示二进制下的 s ，最高位不足 m 则补 0。

【样例 1 输入】

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

【样例 1 输出】

```
011
```


【样例 2】

见下发文件中的 *match/ex_match2.in* 与 *match/ex_match2.ans*。

【样例 3】

见下发文件中的 *match/ex_match3.in* 与 *match/ex_match3.ans*。

【样例 4】

见下发文件中的 *match/ex_match4.in* 与 *match/ex_match4.ans*。

【样例 5】

见下发文件中的 *match/ex_match5.in* 与 *match/ex_match5.ans*。

【测试点约束】

对于全部数据，满足 $1 \leq n, m \leq 10^6$ ， $0 \leq \sum k \leq 10^6$ ， $1 \leq p_1 < p_2 < \cdots < p_k \leq m$ ， $\sum p_k \leq 10^6$ 。

测试点	$n \leq$	$m \leq$
1 ~ 3	2	10^6
4 ~ 6	10	10
7 ~ 8	1000	5
9 ~ 10	5	1000
11 ~ 12	50	50
13 ~ 14	1000	1000
15 ~ 20	10^6	10^6