2024 洛谷网校秋令营提高组
模拟赛

第一试

时间：2024 年 10 月 2 日 14:00 ~ 18:00

题目名称	序列问题	元素迷宫	粒子对撞	树上路径
题目类型	传统型	传统型	传统型	传统型
目录	`sequence`	`element`	`collider`	`path`
可执行文件名	`sequence`	`element`	`collider`	`path`
输入文件名	`sequence.in`	`element.in`	`collider.in`	`path.in`
输出文件名	`sequence.out`	`element.out`	`collider.out`	`path.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB
测试点数目	20	25	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	`sequence.cpp`	`element.cpp`	`collider.cpp`	`path.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14`

序列问题（sequence）

【题目描述】

小 L 给了你 n 个数 $a_1, a_2 \dots a_n$。
你可以进行 m 次操作，每次选择一个 i，将 a_i 变为 2 倍。即：$a_i \leftarrow 2 \times a_i$。
请你求出 m 次操作后，所有数的和的最小值是多少？
由于答案可能很大，你只需要输出答案对 $10^9 + 7$ 取模的结果。

【输入格式】

从文件 ***sequence.in*** 中读入数据。
第一行两个数 n, m。
第二行 n 个数 $a_1, a_2 \dots a_n$。

【输出格式】

输出到文件 ***sequence.out*** 中。
一个整数代表最小值对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

```
3 4
1 2 3
```

【样例 1 输出】

```
14
```

【样例 1 解释】

一种可能的方式是，操作 2 次 a_1 让 a_1 变为 4，操作 1 次 a_2 让 a_2 变为 4，操作 1 次 a_3 让 a_3 变为 6。此时所有数的和的最小值是 14，为可能的最小值。

【样例 2】

见选手目录下的 ***sequence/sequence2.in*** 与 ***sequence/sequence2.ans***。

【样例 3】

见选手目录下的 ***sequence/sequence3.in*** 与 ***sequence/sequence3.ans***。

【样例 4】

见选手目录下的 ***sequence/sequence4.in*** 与 ***sequence/sequence4.ans***。

【样例 5】

见选手目录下的 ***sequence/sequence5.in*** 与 ***sequence/sequence5.ans***。

【样例 6】

见选手目录下的 ***sequence/sequence6.in*** 与 ***sequence/sequence6.ans***。

【样例 7】

见选手目录下的 ***sequence/sequence7.in*** 与 ***sequence/sequence7.ans***。

【子任务】

- 对于 10% 的测试点，$n, m \leq 10$。
- 对于 25% 的测试点，$n, m \leq 100$。
- 对于 40% 的测试点，$1 \leq n, m \leq 5000$。
- 对于 50% 的测试点，$1 \leq m \leq 2 \times 10^5$。
- 对于另外 10% 的测试点，$m \leq 10$。
- 对于另外 10% 的测试点，$n \leq 10$。
- 对于另外 15% 的测试点，a_i 是 2 的次幂。

对于 100% 的测试点，$1 \leq n \leq 2 \times 10^5, 0 \leq m \leq 10^9, 1 \leq a_i \leq 10^9$。

元素迷宫（element）

【题目背景】

本题题面较长，细节较多，请仔细阅读。

【题目描述】

小 L 正在一款游戏中探险“元素迷宫”。玩家会处于一个 $n \times m$ 的迷宫内，玩家可以从一个格子移动到相邻的格子，每个格子有着不同的属性。

- 红色格子（R）
  这些格子是陷阱，玩家进入这些格子会死掉。
- 绿色格子（G）
  每个绿色格子代表了一个秘境副本，玩家需要花费 t_1 秒才能通过这个格子。
- 橙色格子（O）
  当玩家进入橙色格子时，会受到火元素附着，并且需要花费 t_2 秒才能通过这个格子。
- 蓝色格子（B）
  蓝色格子代表水，玩家可以花费 t_3 秒通过这个格子。不过假如玩家当前被附着了火元素，就无法进入蓝色格子。
- 黄色格子（Y）
  黄色格子中存在雷元素方碑，玩家不能进入。特别的，如果该格子旁边是蓝色格子，那么水中就会被附着雷元素，并且会向它旁边的水中也传递雷元素，不过最多传递 d 次（也就是说到该雷元素方碑的曼哈顿距离不超过 d）。当然，被附着了雷元素的水中也不能进入。
  例如：

```
BBBGB
BBBYG
GBBGG
```

  当 $d = 2$ 时，这个例子中，$(1, 5)$ 位置的水因为不和黄色格子联通所以不会被附着雷元素。$(1, 1), (1, 2), (2, 1), (3, 2)$ 位置的水因为和黄色格子曼哈顿距离超过 d 所以也不会被附着。
- 黑色格子（H）
  黑色格子是墙，玩家不能进入。可以认为迷宫外面一圈也是黑色格子。
- 粉色格子（P）
  普通格子，可以花费 t_5 时间通过。
- 紫色格子（V）

当玩家从某方向进入紫色格子时，玩家会沿着进来的方向滑向该方向上的下一个格子，如果还是紫色格子就继续滑行。如果滑到了红色格子，黄色格子或者雷元素附着的蓝色格子，那么玩家就死了。如果滑行方向上下一个是黑色格子，那么玩家不会滑过去，而是会停在当前的紫色格子上。当玩家停到某个紫色格子时，玩家只能通过推旁边的黑色格子借助反作用力滑到反方向的下一个格子。通过一个紫色格子（包括转向）需要花费 t_4 秒。

此外，紫色格子会使玩家附着冰元素，冰元素不会阻止玩家进入水中。这里玩家只能同时被附着最多一种元素，后附着的元素会覆盖先附着的。

这是一个例子：

```
RRVR
RRVR
PVVH
HHHH
```

当玩家从 $(1,3)$ 向下滑时，会被 $(4,3)$ 位置的黑色格子挡住，此时玩家可以推 $(3,4)$ 位置的黑色格子从而向左滑去，到达 $(3,1)$。

不过如果变成下面的样子

```
RRVR
RRVR
PVVG
HHHH
```

就会因为无法推黑色格子所以不能到达 $(3,1)$。

请求出，玩家从 (S_x,S_y) 移动到 (T_x,T_y) 最少需要花费多少时间。这里保证起点和终点不同，且都是粉色格子。特别的，如果不能到达，输出 -1。

【输入格式】

从文件 ***element.in*** 中读入数据。

第一行 6 个整数 n,m,S_x,S_y,T_x,T_y，表示地图大小和 S,T 坐标。

第二行 6 个整数 t_1,t_2,t_3,t_4,t_5,d。

接下来 n 行，每行 m 个字符的字符串，表示迷宫地图。

【输出格式】

输出到文件 ***element.out*** 中。

一个整数表示答案。

【样例 1 输入】

```
5 5 1 1 5 5
1 1 4 5 1 4
PHPPP
PHPHP
PHPHP
PHPHP
PPPHP
```

【样例 1 输出】

```
16
```

【样例 2 输入】

```
10 10 1 1 8 1
2 1 1 1 1 2
PBBYBBBBBY
BBBBBBBBBB
BBBBBBBBBB
BBBBBBBBBB
YBBGBBYBBB
BBGGGBBBBB
BGGGGGBBBB
PBGGGBBBBB
BBBGBBBBBY
BBBBBBBBBB
```

【样例 2 输出】

```
18
```

【样例 3 输入】

```
5 5 1 1 3 3
1 1 1 1 1 1
PPHPH
VHVVV
VHPHV
VVHVV
HVVVH
```

【样例 3 输出】

```
14
```

【样例 4 输入】

```
5 5 1 1 3 3
1 1 1 1 1 1
PPHPV
VHVVV
VHPHV
VVHVV
HVVVH
```

【样例 4 输出】

```
-1
```

【样例 5 输入】

```
3 3 1 1 3 3
1 1 1 1 1 1
POB
OOB
BBP
```

【样例 5 输出】

```
-1
```

【样例 6 输入】

```
3 3 1 1 3 3
1 1 1 1 1 1
PHH
VYH
VBP
```

【样例 6 输出】

```
-1
```

【样例 7 输入】

```
3 3 1 1 3 3
1 1 10 1 2 0
PHH
VYH
OBP
```

【样例 7 输出】

```
-1
```

【样例 8 输入】

```
5 5 1 1 5 5
8 2 4 1 2 5
PBVPV
PPBOH
HVVBB
VOHOG
```

```
OPPBP
```

【样例 8 输出】

```
-1
```

【样例 9 输入】

```
10 10 1 1 10 10
1 2 3 4 5 2
PRYGBGPVBR
VRHVPPOOVG
PBPVGVVYBV
OOVPOOBOOR
VBRORHOYRH
OVYPOVHGHO
GVRPVPPGYO
GHHORVGRVR
VPRGVVROVH
RROGROGGPP
```

【样例 9 输出】

```
62
```

【子任务】

本题共有 25 个测试点。对于所有测试点，满足：$1 \leq n \leq 500, 1 \leq m \leq 500, 1 \leq t_i \leq 500 (i \in \{1, 2, 3, 4, 5\}), 1 \leq d \leq 500$。

对于每个测试点，分别有如下的限制：

测试点编号	$n \leq$	$m \leq$	$t_i \leq$	$d \leq$	特殊性质
$1, 2, 3$	10	10	10	10	
$4, 5, 6$	50	50	50	50	
$7, 8, 9, 10, 11$	500	500	500	500	没有橙色格子（O）
$12, 13, 14, 15$	500	500	500	500	没有紫色格子（V）
$16, 17, 18$	500	500	500	200	只有黑色格子（H）和粉色格子（P）
$19, 20$	200	200	200	200	
$21, 22, 23, 24, 25$	500	500	500	500	

粒子对撞（collider）

【题目描述】

小 L 在一个横向加速器里从左到右依次放置了 n 个粒子，第 i 个粒子被设置成向左或者向右发射。

当两个粒子相撞时，有 $\frac{x}{y}$ 的概率左边的粒子被撞碎，有 $1-\frac{x}{y}$ 的概率右边的粒子被撞碎，剩下的那个粒子会沿着原方向继续移动。

设最终有 a 个粒子从加速器左端射出，有 b 个粒子从加速器右端射出的概率为 $p_{a,b}$，请求出下面式子的值：

$$\sum_{a=0}^{n}\sum_{b=0}^{n}p_{a,b}f_ag_b$$

其中 f,g 是两个输入的数组。

请求出该值对 10^9+7 取模的结果。

【输入格式】

从文件 ***collider.in*** 中读入数据。

第一行三个整数 n,x,y。

第二行 $n+1$ 个整数代表 $f_0,f_1,\dots f_n$。

第三行 $n+1$ 个整数代表 $g_0,g_1,\dots g_n$。

第四行一个长度为 n 的，只包含 `L` 和 `R` 的字符串，分别代表第 i 个粒子被设置成向左还是向右。

【输出格式】

输出到文件 ***collider.out*** 中。

一个整数代表要求的值。

【样例 1 输入】

```
3 1 2
1 0 1 0
1 1 0 0
RRL
```

【样例 1 输出】

```
250000002
```

【样例 1 解释】

有三种情况：

- 只有粒子 3 从左边射出，概率为 $p_{1,0}=\frac{1}{4}$。
- 只有粒子 1 从右边射出，概率为 $p_{0,1}=\frac{1}{4}$。
- 粒子 $1,2$ 都从右边射出，概率为 $p_{0,2}=\frac{1}{2}$。

因此答案为 $p_{1,0}\times 0\times 1+p_{0,1}\times 1\times 1+p_{0,2}\times 1\times 0=\frac{1}{4}\equiv 250000002 \pmod{10^9+7}$。

【样例 2 输入】

```
4 53453572 93220202
55002773 15903962 72731481 44921378 29816614
68694414 92169661 34294667 10409058 86703386
LRLL
```

【样例 2 输出】

```
169984330
```

【样例 3】

见选手目录下的 ***collider/collider3.in*** 与 ***collider/collider3.ans***。

【样例 4】

见选手目录下的 ***collider/collider4.in*** 与 ***collider/collider4.ans***。

【样例 5】

见选手目录下的 ***collider/collider5.in*** 与 ***collider/collider5.ans***。

【样例 6】

见选手目录下的 ***collider/collider6.in*** 与 ***collider/collider6.ans***。

【样例 7】

见选手目录下的 ***collider/collider7.in*** 与 ***collider/collider7.ans***。

【样例 8】

见选手目录下的 ***collider/collider8.in*** 与 ***collider/collider8.ans***。

【子任务】

- 对于 15% 的数据，$n \leq 10$。
- 对于 30% 的数据，$n \leq 20$。
- 对于 45% 的数据，$n \leq 50$。
- 对于 60% 的数据，$n \leq 800$。
- 对于另外 20% 的数据，s_i 里 `R` 构成了一个前缀。

对于 100% 的数据，$1 \leq n \leq 5000, 0 \leq f_i, g_i \leq 10^8, 1 \leq x \leq y \leq 10^8$。

【提示】

对于有理数 $c = \frac{a}{b}$，c 对 p（p 是一个质数）取模的值被定义为关于 x 的同余方程 $bx \equiv a \pmod{p}$ 的解。

树上路径（path）

【题目描述】

小 L 给了你一棵 n 个点的无根树，边有权值。

小 L 希望你找到若干条简单路径（可以不选），满足每条简单路径都恰好包含了 k 条边，且每条边最多在一条路径里。

请你最大化路径包含的边的权值和。

【输入格式】

从文件 ***path.in*** 中读入数据。

第一行两个数 n, k。

接下来 $n-1$ 行，每行三个数 u_i, v_i, w_i 代表一条连接了 u_i, v_i，权值为 w_i 的边。

【输出格式】

输出到文件 ***path.out*** 中。

一个整数代表最大的权值。

【样例 1 输入】

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

【样例 1 输出】

```
1
```

【样例 2】

见选手目录下的 ***path/path2.in*** 与 ***path/path2.ans***。

【样例 3】

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

【样例 4】

见选手目录下的 ***path/path4.in*** 与 ***path/path4.ans***。

【样例 5】

见选手目录下的 ***path/path5.in*** 与 ***path/path5.ans***。

【样例 6】

见选手目录下的 ***path/path6.in*** 与 ***path/path6.ans***。

【样例 7】

见选手目录下的 ***path/path7.in*** 与 ***path/path7.ans***。

【子任务】

- 对于其中 10% 的数据，$1 \leq n \leq 16$。
- 对于 20% 的数据，$1 \leq n \leq 500$。
- 对于另外 15% 的数据，$1 \leq n \leq 5000, k = 3$。
- 对于另外 15% 的数据，$1 \leq n \leq 5000, k = 4$。
- 对于另外 20% 的数据，$k = 3$。

对于 100% 的数据，$1 \leq n \leq 2 \times 10^5, k \in \{3, 4\}, -10^9 \leq w_i \leq 10^9$。