

# 云斗 CSP-S 模拟赛

时间：2025 年 10 月

题目名称	修复括号串	序列	海棠花落	停车场
题目类型	传统型	传统型	传统型	传统型
可执行文件名	fix	seq	sakura	park
时间限制	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	256 MiB	256 MiB	1024 MiB	1024 MiB
测试点数目	10	20	30	20
测试点等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	fix.cpp	seq.cpp	sakura.cpp	park.cpp
-----------	---------	---------	------------	----------

编译选项

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

注意事项：

1. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
2. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
3. 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。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
6. 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。
7. 线上提交渠道：[比赛链接](#)。注意，线上提交也应采用文件输入输出。



## 修复括号串 (fix)

### 【题目描述】

给定一个长度为  $n$  ( $n$  为偶数), 由左右括号组成的字符串  $S$ 。

每次操作可以修改  $S$  中的某个字符, 求出最少的操作次数, 使  $S$  变成一个括号串。

我们定义满足以下条件的字符串为括号串。

1. 空串是括号串。
2. 如果  $A$  是括号串,  $(A)$  是括号串。
3. 如果  $A, B$  都是括号串,  $AB$  是括号串。

### 【输入格式】

第一行输入一个整数  $n$ 。

第二行输入一个长度为  $n$  的字符串  $S$ ,  $S$  中每个字符为左括号或者右括号。

### 【输出格式】

输出一行一个整数, 表示最少的操作次数。

### 【样例输入 1】

```
1 4
2 ())(
```

### 【样例输出 1】

```
1 2
```

### 【样例输入 2】

```
1 2
2 ((
```

### 【样例输出 2】

```
1 1
```

**【数据范围和约定】**

对于 30% 的数据,  $n \leq 10$  。

对于另外 30% 的数据,  $n \leq 10^3$  。

对于另外 20% 的数据,  $S$  中仅包含左括号。

对于全部数据,  $1 \leq n \leq 10^6$ ,  $n$  是偶数,  $S$  长度为  $n$ , 而且每个字符为左括号或者右括号。

## 序列 (seq)

### 【题目描述】

给定一个长度为  $n$  的整数数列  $a_1, a_2, \dots, a_n$ 。cmx 需要执行若干次操作，每次操作选定一个整数  $x$  ( $1 \leq x \leq n$ ) 和一个整数  $k$ ，然后令  $a_x \leftarrow k$ ，使得所有操作结束后：

1. 整个数列是回文的，即  $\forall i \in \{1, 2, \dots, n\}, a_i = a_{n-i+1}$ ；
2. 数列的前一半是严格递增的，即  $\forall i \in \{1, 2, \dots, \lceil n/2 \rceil - 1\}, a_i < a_{i+1}$ 。

求最小操作次数。

### 【输入格式】

第一行一个正整数  $n$ ，表示数列长度；

第二行  $n$  个非负整数表示给定数列。

本题读入规模较大，请使用效率较高的读入方式。

### 【输出格式】

一个非负整数表示答案。

### 【样例输入 1】

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

### 【样例输出 1】

```
1 0
```

### 【样例输入 2】

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

### 【样例输出 2】

```
1 1
```

**【数据范围和约定】**

对于全部数据,  $1 \leq n \leq 5 \cdot 10^5$ ,  $0 \leq a_i \leq 10^9$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 4	20	否
5 ~ 6	5000	是
7 ~ 14	5000	否
15 ~ 16	$5 \cdot 10^5$	是
17 ~ 20	$5 \cdot 10^5$	否

特殊性质: 答案小于等于 2。

## 海棠花落 (sakura)

### 【题目描述】

苦竹岭无归去日，海棠花落旧栖枝。

春风三月，诗岸独自一人离家求学。在西北的某座城市里，花开正好。

这颗樱花树对诗岸来说，是一颗  $n$  个点的树，其中第  $i$  个点上的开的花有观察值  $a_i$ ，这代表对诗岸来说看到这朵花，会让她增加  $a_i$  的悲伤度。

诗岸将在树前站立  $10^{2025}$  秒，其中每一秒中会发生两件事：

首先，每一朵这一秒仍然存在于树上的花会对诗岸产生悲伤度，使得诗岸观察花朵的悲伤度总和增加这些花朵的观察值  $a_i$  之和。

接着，诗岸将采摘一些花朵，被采摘的花朵将离开树上，从而无法在后续的观测中影响诗岸。但由于诗岸对于樱花树整体美观的考量，诗岸每次采摘的花朵直接两两不能有连边，这保证了樱花树每次都不会被采摘太多从而导致局部空白。

现在诗岸想问，在这漫长的时间内，她所产生的悲伤度总和最少是多少。

形式化表达：

给定一颗树  $T$ ，你需要选出若干独立子集(可为空)  $S_1, S_2, \dots, S_{10^{2025}}$  其中  $S_i \cap S_j = \emptyset$ ,  $\bigcup_{i=1}^{10^{2025}} S_i = T$

最小化  $\sum_{i=1}^{10^{2025}} \sum_{j=i}^{10^{2025}} \sum_{x \in S_j} a_x$ 。

### 【输入格式】

在所有输入前将输入  $id$ ，表示这一数据的测试编号，详情见数据范围。

数据第一行输入  $n$ ，代表位于诗岸面前的樱花树的大小。

数据第二行输入  $n$  个数，其中第  $i$  个数代表  $a_i$ 。

接下来  $n - 1$  行，每行输入一个数  $f_i$ ，代表在树上存在  $(i + 1, f_i)$  这一条边，树保证以 1 为根。

同时请选手注意最后两档数据中的特殊输入格式。

### 【输出格式】

输出一行数字，表示诗岸在漫长时光里能收到的最小的悲伤总和。

## 【数据范围】

测试编号	$n \leq$	$a_i \leq$	特殊性质
1 ~ 3	10	100	无
4 ~ 8	$4 \times 10^3$	$10^8$	无
9 ~ 18	$3 \times 10^5$	$10^8$	无
19 ~ 20	$10^6$	$10^8$	给出的树保证每个点的度数小于等于 5, 同时 $n = 10^6$
21 ~ 30	$10^6$	$10^8$	$n = 10^6$

其中对于 19 至 30 组数据, 由于输入过大, 我们将给出数据的 *gen* 和 *seed*  
其中 *gen* 为

```

1  int data_f[1000005], data_a[1000005];
2  long long r(){return (rand()<<15) + rand();}
3  inline void generate_deg(int seed){
4      //....
5  }
6  inline void generate(int seed){
7      //....
8  }
9
10 int main(){
11     //id
12     //if id in [19,20] use generate_deg
13     //if id in [21,30] use generate
14 }
```

具体实现可以见下发文件 *gen.cpp*, 但保证本题做法与数据生成器无关。

在  $id = 19/20$  时, 通过输入的 *seed* 调用 **generate\_deg** 函数

在  $id = [21, 30]$  时, 通过输入的 *seed* 调用 **generate** 函数

输入数据将为对应  $data\_a_{1..n}$  与  $data\_f_{1..n-1}$

为方便选手我们提供了一个现成框架, 可供选手参考, 可以查看下发文件 *exp.cpp*

```

1  int data_f[1000005], data_a[1000005];
2  long long r(){return (rand()<<15) + rand();}
3  inline void generate_deg(int seed){
4      //....
5  }
6  inline void generate(int seed){
```

```
7      //....
8  }
9
10 int n;
11 int seed;
12
13 int main(){
14     scanf("%d",&id);
15     if(id < 19){
16         scanf("%d",&n);
17         for(int i = 1;i <= n;++i)scanf("%d",&data_a[i]);
18         for(int i = 1;i < n;++i)scanf("%d",&data_f[i]);
19     }
20     if(id == 19 || id == 20){n = 1000000;scanf("%d",&seed);
        generate_deg(seed);}
21     if(id > 20){n = 1000000;scanf("%d",&seed);generate(seed);}
22 }
```



## 停车场 (park)

### 【题目描述】

给定一张  $n \times m$  的网格图，每个格子要么是一块空地 (用 o 表示, ASCII=111)，要么是一辆车 (用 x 表示, ASCII=120)，要么是一个出口 (用 e 表示, ASCII=101)。

每辆车都可以向上下左右四个方向移动，且只能移动到空地或者出口上。

称一辆车可以移动到出口，需满足：整张网格上只有这一辆车可以移动，且在不碰到任何其他车的情况下（只能移动到空地或者出口上）移动到某一出口。

称一张网格图是完美的，需满足：对于网格上的每一辆车，都满足该车可以移动到出口。

搓完一天的雀魂麻将，lindongli2004 和 Zimse 已经没有脑子思考，他们给了你一张完美的网格图，并想让你帮他们求出满足下列条件的空地个数：如果在该空地上放上一辆车，这张网格图依旧完美。

### 【输入格式】

第一行一个正整数  $T$  表示数据组数。

对于每组数据：

第一行两个正整数  $n, m$  表示网格图的行数和列数。

接下来  $n$  行每行一个长度为  $m$  的字符串，描述该网格图。

### 【输出格式】

输出  $T$  行，每行一个整数表示该组数据的答案。

### 【样例输入 1】

```
1 2
2 3 3
3 xex
4 ooo
5 xox
6 2 4
7 eoxo
8 xoox
```

**【样例输出 1】**

```
1 3
2 0
```

**【数据范围和约定】**

记  $S = \sum n \times m$ ，即全部  $T$  组数据的网格图大小之和， $ex$  为网格图中出口的数量， $x$  为网格图中车的数量。

对于前 10% 的数据，满足  $S \leq 12$ 。

对于另外 10% 的数据，满足  $S \leq 500$ 。

对于另外 10% 的数据，满足  $S \leq 5000$ 。

对于另外 10% 的数据，满足  $n = 1$ 。

对于另外 10% 的数据，满足  $x \leq 20$ 。

对于另外 10% 的数据，满足  $n = 2, ex = 1$ 。

对于 100% 的数据，满足  $S \leq 10^6, 1 \leq n, m \leq 10^6$ 。