

# 华师一 NOIP2020 模拟测试

2020 年 11 月 29 日 8:00~12:30

## 一. 题目概况

中文题目名称	括号匹配	成绩排名	字符串距离	树上染色
英文题目名称	bracket	rank	distance	color
主文件名	bracket	rank	distance	color
输入文件名	bracket.in	rank.in	distance.in	color.in
输出文件名	bracket.out	rank.out	distance.out	color.out
每个测试点时限	1 秒	1 秒	1 秒	2 秒
内存上限	256MB	256MB	256MB	512MB
测试点数目	10	10	20	20
每个测试点分值	10	10	5	5
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统

## 二. 提交源程序文件名

对于 C++ 语言	bracket.cpp	rank.cpp	distance.cpp	color.cpp
-----------	-------------	----------	--------------	-----------

## 三. 编译命令（对于 C++语言）

题目	C++ 语言编译命令
bracket	-O2 -std=c++11
rank	-O2 -std=c++11
distance	-O2 -std=c++11
color	-O2 -std=c++11 -Wl,--stack=268435456

# 括号匹配

## 【题目描述】

给定一字符串，只由"(", ")", "\*"三种字符组成，你需要替换掉所有的"\*"号使得整个字符串变为一个平衡的括号串。你可以选择将一个"\*"替换为一个"("或")"，也可以直接删掉这个"\*"（也就是替换为一个空串）。

一个平衡的括号串定义如下：

- (1) 空串是平衡的括号串。
- (2) 如果A是一个平衡的括号串，那么(A)也是一个平衡的括号串。
- (3) 如果A、B都是平衡的括号串，那么AB也是一个平衡的括号串（这里AB指将A与B拼接起来）。

例如："()()()"、"((()))"、"()()()()"以及""这些都是平衡的括号串，")()()"、"(((())"、"())"这些不是平衡的括号串。

给定这样一个含有"\*"的括号串，输出替换掉所有"\*"后得到的一个平衡的括号串。由于可能存在多种答案，所以本题只要求选手输出长度最短且字典序最小的括号串（即所有长度最短的答案中字典序最小的一个）。如果不能得到一个平衡的括号串，输出"No solution!"。

如果有两个长度均为 $n$ 的字符串 $A$ 和 $B$ ，假设存在一个整数 $k$ 满足 $1 \leq k \leq n$ ， $A$ 和 $B$ 的前 $k - 1$ 个字符完全相等，且 $A$ 的第 $k$ 个字符小于 $B$ ，我们就称 $A$ 的字典序小于 $B$ 。例如，"()()"的字典序小于"()())"的字典序。

## 【输入】

输入仅一行，即一个字符串，只含有"(", ")", "\*"三种字符。

## 【输出】

输出一行，即替换掉所有"\*"后长度最短且字典序最小的平衡括号串。如果无解，输出"No solution!"。

## 【样例输入1】

```
((***)()((**
```

## 【样例输出1】

```
((()))()((())
```

## 【样例输入2】

```
()(**)()(
```

## 【样例输出2】

```
No solution!
```

## 【样例说明】

对于第一组样例，可以将前3个"\*"替换为空串，后两个"\*"替换为")"，这样便得到一个长度最短且字典序最小的答案"()()()()"。

对于第二组样例，无论怎样进行替换都无法得到一个平衡的括号串，故输出"No solution!"。

## 【数据范围】

设字符串长度为 $n$ ：

对于20%的数据，有 $1 \leq n \leq 15$

对于40%的数据，有 $1 \leq n \leq 10^3$

对于100%的数据，有 $1 \leq n \leq 10^5$

# 成绩排名

## 【题目描述】

小Z毕业后去了H中学教书，他带的班级有 $n$ 个学生，对于每个学生 $i$ 可以用一个正整数 $A_i$ 来衡量其学习能力（ $i = 1, 2, \dots, n$ ）。有一天，小Z捡到了 $k$ 本神奇教材，如果给一个学生学习了这本教材，他的学习能力就会变成原来的 $t$ 倍。小Z决定将这 $k$ 本书随机且等概率地发放给班上的 $n$ 个同学，每个学生最多只能得到一本书，也就是说班上会有 $k$ 名同学被分配到书，学习能力变成 $t$ 倍，而剩下 $n - k$ 名同学没有书，学习能力不变。现在小Z想知道，在所有的分配方案中，对于班上的每个同学来说排名保持不变的情况分别有多少种。

在这里，排名是指学习能力比他高的人的数目加一，比方说有5个同学，学习能力为[3, 2, 2, 1, 1]，他们的排名即[1, 2, 2, 4, 4]。假设小Z捡到了2本能使学习能力翻4倍的神奇教材，他将这2本书交给了第2个和第5个同学，那么现在这五个人的学习能力变为[3, 8, 2, 1, 4]，排名也就变为[3, 1, 4, 5, 2]。

## 【输入】

第一行输入3个整数 $n, k, t$ ，分别代表学生的数目、神奇教材的数目以及学习能力翻的倍数。第二行输入 $n$ 个整数 $A_i$ ，代表每个学生的学习能力，编号从1到 $n$ 。

## 【输出】

输出 $n$ 行，第 $i$ 行输出一个整数代表第 $i$ 位同学在Z老师发完书后学习能力排名保持不变的所有可能的情况数目。由于这些数可能非常大，请输出其对 $10^9 + 7$ 取模后的结果。

## 【样例输入】

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

## 【样例输出】

```
4
3
2
4
```

## 【样例说明】

为方便说明，这里用 $\{x, y\}$ 表示把书发放给学生 $x$ 和学生 $y$ 。

对于学生1来说，可能的方案有 $\{1, 2\}$ ， $\{2, 3\}$ ， $\{3, 4\}$ ， $\{2, 4\}$ 。下面逐个进行详细解释：

- 1.  $\{1, 2\}$ ，学习能力变为[2, 4, 3, 4]，排名变为[4, 1, 3, 1]，学生1的排名保持不变。
- 2.  $\{2, 3\}$ ，学习能力变为[1, 3, 6, 4]，排名变为[4, 3, 1, 2]，学生1的排名保持不变。
- 3.  $\{3, 4\}$ ，学习能力变为[1, 2, 6, 8]，排名变为[4, 3, 2, 1]，学生1的排名保持不变。
- 4.  $\{2, 4\}$ ，学习能力变为[1, 4, 3, 8]，排名变为[4, 2, 3, 1]，学生1的排名保持不变。

其他学生同理可得类似结果，这里只给出方案不再进行详细解释。

对于学生2来说，可能的方案有 $\{1, 3\}$ ， $\{3, 4\}$ ， $\{1, 4\}$ 。

对于学生3来说，可能的方案有 $\{3, 4\}$ ， $\{1, 4\}$ 。

对于学生4来说，可能的方案有 $\{1, 2\}$ ， $\{1, 4\}$ ， $\{2, 4\}$ ， $\{3, 4\}$ 。

### 【数据范围】

对于10%的数据，有 $1 \leq n \leq 10$

对于30%的数据，有 $1 \leq n \leq 10^3$

对于另外10%的数据，有 $1 \leq n \leq 10^5$ 且 $k = 1$

对于100%的数据，有 $1 \leq n \leq 10^5$ ,  $1 \leq k \leq n$ ,  $2 \leq t \leq 1000$ ,  $1 \leq A_i \leq 10^9$

# 字符串距离

## 【题目描述】

对于两个小写字母组成的字符串 $S$ 和 $T$ ，规定两种操作如下。

- 操作一：选择 $S$ 或 $T$ 上任意一个字符并删除这个字符。例如，对于"aab"和"cdd"，可以选择第一个串的第三个字符'b'删除，得到"aa"，第二个串保持不变。
- 操作二：选择 $S$ 或 $T$ 上任意一个位置，在这个位置上插入任意一个小写字母（也可以选择字符串的开头或结尾）。例如，对于"aab"和"cdd"，可以选择第二个串的'c'与'd'中间的位置并插入一个'e'，得到"cedd"，第一个串保持不变。

规定两字符串之间的**距离**为使得两字符串变得完全相同所需要的最少操作步骤数，对于每一个步骤都可以任选上述两种操作的其中一个执行。可以证明，一定可以通过执行有限次上述操作，使得两个字符串变得完全相同。

现给定两个字符串 $A$ 和 $B$ ，以及 $q$ 次询问，每次询问以两个整数 $l$ 和 $r$ 的形式给出，你需要求出字符串 $A[l \dots r]$ （即字符串 $A$ 的区间 $[l, r]$ 上的字符组成的子串）和字符串 $B$ 的距离是多少。

## 【输入】

第一行输入一个字符串 $A$ ，第二行输入一个字符串 $B$ ，第三行输入一个正整数 $q$ ，表示询问的次数。

接下来 $q$ 行，每次输入两个整数 $l$ 和 $r$ ，代表区间 $[l, r]$ 。

## 【输出】

对于每次询问，输出一个正整数，表示字符串 $A[l \dots r]$ 和字符串 $B$ 的距离。

## 【样例输入】

```
qaqaqwqaqaq
qaqwqaq
3
1 7
2 8
3 9
```

## 【样例输出】

```
4
2
0
```

## 【样例说明】

对于第一次询问 $[1, 7]$ ，也就是询问字符串"qaqaqwq"和字符串"qaqwqaq"的距离，可以选择在第一个字符串的最后一个字符后依次插入'a'和'q'，然后在第二个字符串的第一个字符前依次插入'a'和'q'，使得两个字符串都变为"qaqaqwqaq"，需要4次操作，距离为4

对于第二次询问 $[2, 8]$ ，也就是询问字符串"aqaqwqa"和字符串"qaqwqaq"的距离，可以选择删除第一个字符串的第一个字符'a'，然后在第一个字符串的最后一个字符后面插入一个'q'，使得两个字符串都变为"qaqwqaq"，需要2次操作，距离为2

对于第三次询问[3, 9]，也就是询问字符串"qaqwqaq"和字符串"qaqwqaq"的距离，两字符串已经完全相同，不需要再进行操作，距离为0

### 【数据范围】

对于40%的数据，有 $1 \leq n \leq 500$ ， $1 \leq q \leq 10^3$

对于100%的数据，有 $1 \leq n \leq 10^5$ ， $1 \leq q \leq 10^5$

# 树上染色

## 【题目描述】

现有一颗含有 $n$ 个结点的有根树，根固定为标号为1的结点。对于树上每个结点 $i$  ( $i = 1, 2, \dots, n$ )都有一个权值 $val_i$ 和一种颜色 $col_i$ ，颜色标号从1到 $n$ ，代表 $n$ 种颜色。

现有两种操作如下：

- 1 x v 将结点 $x$ 的权值设为 $v$
- 2 x c 将结点 $x$ 的颜色设为 $c$

对于所有满足 $1 \leq u < v \leq n$ 且不构成祖先关系（即 $u$ 不是 $v$ 的祖先结点，且 $v$ 不是 $u$ 的祖先结点）的一对结点 $u, v$ ，规定 $f(u, v)$ 的值如下，其中 $\oplus$ 是按位异或符号：

$$f(u, v) = \begin{cases} val_u \oplus val_v, & col_u = col_v \\ 0, & col_u \neq col_v \end{cases}$$

规定 $S$ 的值为：

$$S = \sum_{\substack{1 \leq u < v \leq n \\ u \text{ 和 } v \text{ 不构成祖先关系}}} f(u, v)$$

给定 $q$ 个操作，要求选手求出对于第 $i$ 次 ( $i = 0, 1, 2, \dots, q$ ) 操作后 $S$ 的值（第0次的结果也就是在所有操作开始前的初始状态下 $S$ 的值）。

## 【输入】

第一行输入一个正整数 $n$ 表示结点个数。

第二行输入 $n$ 个正整数，表示结点 $i$ 的初始颜色 $col_i$ 。

第三行输入 $n$ 个正整数，表示结点 $i$ 的初始权值 $val_i$ 。

接下来的 $n - 1$ 行，每行输入两个正整数 $u$ 和 $v$  ( $1 \leq u, v \leq n$ )，表示树上一条连接了结点 $u$ 和结点 $v$ 的边。题目保证所给数据可以组成一棵合法的树。

接下来一行输入一个正整数 $q$ 表示操作的次数。

接下来的 $q$ 行，每行输入3个整数，如果第一个整数的值为1，则代表第一种操作，那么后两个数 $x$ 和 $v$ 就表示将结点 $x$ 的权值设为 $v$ ；如果第一个整数的值为2，则代表第二种操作，那么后两个数 $x$ 和 $c$ 就表示将结点 $x$ 的颜色设为 $c$ 。

请选手注意：数据量较大，请尽量使用快速读入方式来读入数据。这里提供一段快速读入的代码供选手参考。

```
int read() {
    int num=0, f=1;
    char c=getchar();
    while(c<'0' || c>'9') {
        if(c=='-') f=-1;
        c=getchar();
    }
    while(c>='0' && c<='9') num=num*10+c-'0', c=getchar();
    return num*f;
}
```



## 【输出】

输出 $q + 1$ 行，每行表示第 $i$ 次( $i = 0, 1, 2, \dots, n$ )操作后 $S$ 的值。

## 【样例输入】

```
5
1 1 1 1 1
1 2 4 8 16
1 2
3 1
2 4
2 5
2
1 3 32
2 3 2
```

## 【样例输出】

```
62
146
24
```

## 【样例说明】

根据树的结构容易得到：不构成祖先关系的点对有 $\langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 3, 5 \rangle, \langle 4, 5 \rangle$

初始状态，所有结点的颜色均为1，其中 $f(2, 3) = 6, f(3, 4) = 12, f(3, 5) = 20, f(4, 5) = 24$ ，这些数求和便得到62

第一次操作，将点3的权值修改为32，得到 $f(2, 3) = 34, f(3, 4) = 40, f(3, 5) = 48, f(4, 5) = 24$ ，求和得到146

第二次操作，将点3的颜色修改为2，现在点1, 2, 4, 5的颜色相同，而点3与其他点的颜色都不同，容易得到 $f(2, 3) = 0, f(3, 4) = 0, f(3, 5) = 0, f(4, 5) = 24$ ，求和得到24

## 【数据范围】

对于前10%的数据，有 $1 \leq n \leq 100$

对于前20%的数据，有 $1 \leq n \leq 1000, 1 \leq q \leq 100$

对于前30%的数据，有 $1 \leq n \leq 1000, 1 \leq q \leq 1000$

对于前60%的数据，有 $1 \leq n \leq 10^5, 1 \leq q \leq 20000, 0 \leq val_i < 2^{15}$

对于全部的数据，有 $1 \leq n \leq 10^5, 1 \leq q \leq 10^5, 1 \leq col_i \leq n, 0 \leq val_i < 2^{20}$