

五校联考

GIEI PION 3202 寻找模式

题目名称	括号	跳跃	集了个合	数树
题目类型	传统型	传统型	传统型	传统型
目录	bracket	jump	set	tree
可执行文件名	bracket	jump	set	tree
输入文件名	bracket.in	jump.in	set.in	tree.in
输出文件名	bracket.out	jump.out	set.out	tree.out
每个测试点时限	1000 ms	3000 ms	1000 ms	1000 ms
内存限制	512 MiB	1024 MiB	128 MiB	512 MiB
子任务数目	6	4	10	5

提交源程序文件名

对于 C++ 语言	bracket.cpp	jump.cpp	set.cpp	tree.cpp
-----------	-------------	----------	---------	----------

编译选项

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

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 选手提交的程序源文件必须不大于 100KB。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 提交方式、评测环境以及是否使用文件输入输出请以各校实际评测环境为准。

#1

传统	1000 ms	512 MiB
----	---------	---------

括号 (bracket)

【题目描述】

给定一个长为 n 的括号串，其中有一些是 ?。给定常数 a, b 。你要取出其中若干不交的子段，使得每一段都存在把 ? 替换成括号的方法使之成为合法括号串。长度为 i 的段的收益 $ai + b$ ，定义一个方案的收益为其所有段的收益和。求最大可能的收益。

【输入格式】

第一行输入三个整数 n , a 和 b 。第二行输入给定括号串。

【输出格式】

输出一行，包含一个整数，表示答案。

【样例数据】

【样例 1 输入】

```
10 -406834 918906
?)?) (?((()
```

【样例 1 输出】

```
420952
```

【样例 2】

见下发文件 ex_bracket2.in 和 ex_bracket2.out，该测试用例满足子任务 2 的约束条件。

【样例 3】

见下发文件 ex_bracket3.in 和 ex_bracket3.out，该测试用例满足子任务 3 的约束条件。

【样例 4】

见下发文件 ex_bracket4.in 和 ex_bracket4.out，该测试用例满足子任务 4 的约束条件。

【样例 5】

见下发文件 ex_bracket5.in 和 ex_bracket5.out。

【数据规模与约定】

全部数据均保证 $1 \leq n \leq 5 \times 10^5$, $|a|, |b| \leq 10^6$, 输入的字符串只包含 ()? 三种字符。

本题采用子任务捆绑测试, 各个子任务详细信息如下:

编号	子任务分数	$n \leq$	特殊性质
1	5	10	
2	15	500	
3	20	5×10^3	
4	5	5×10^5	输入的字符串只包含 ? 一种字符
5	5	5×10^5	输入的字符串只包含 () 两种字符
6	50	5×10^5	

#2

传统	3000 ms	1024 MiB
----	---------	----------

跳跃 (jump)

【题目描述】

给出一个 $1, \dots, n+1$ 的排列 $v_{1, \dots, n+1}$, 与两组权值 $a_{1, \dots, n}$ 、 $b_{1, \dots, n}$ 。满足 $v_{n+1} = n+1$ 。构造一张 $n+1$ 个点的 **有向图**:

- 对于 $i = 1, \dots, n$, 从 i 向 $i+1$ 连一条权值为 a_i 的边;
- 对于 $i = 1, \dots, n$, 找到最小的 $i < j \leq n+1$ 满足 $v_j > v_i$, 从 i 向 j 连一条权值为 b_i 的边。

定义一条路径的权值为路径上边权的 **最大值**。特别地, 若一条路径不包含任何边, 则其权值为 0。

现在有 q 次询问, 每次询问给出 x, y ($x \leq y$), 求 x 到 y 的权值最小的路径的权值。

【输入格式】

输入的第一行包含三个数 n, q, S , 其中 S 表示测试点所属的子任务编号。第二行包含 n 个数, 表示 $v_{1, \dots, n}$ 。第三行包含 n 个数, 表示 $a_{1, \dots, n}$ 。第四行包含 n 个数, 表示 $b_{1, \dots, n}$ 。

接下来 q 行, 其中第 i 行包含两个数 x, y , 表示第 i 次询问。

【输出格式】

输出应包含 q 行, 第 i 行包含一个整数, 表示你对第 i 个询问求出的答案。

【样例数据】

【样例 1 输入】

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

```
6 7  
3 3
```

【样例 1 输出】

```
2  
3  
3  
5  
3  
0
```

【样例 2】

见下发文件 ex_jump2.in 和 ex_jump2.out，该测试用例满足子任务 1 的约束条件。

【样例 3】

见下发文件 ex_jump3.in 和 ex_jump3.out，该测试用例满足子任务 1 的约束条件。

【样例 4 输入】

```
8 10 2  
8 7 6 5 4 3 2 1  
5 2 3 5 9 4 3 8  
7 3 4 4 1 2 8 3  
1 3  
6 8  
7 9  
9 9  
1 9  
1 8  
4 4  
2 9  
2 6  
1 3
```

【样例 4 输出】

```
5  
4  
3
```

```
0  
5  
9  
0  
3  
9  
5
```

【样例 5】

见下发文件 ex_jump5.in 和 ex_jump5.out，该测试用例满足子任务 2 的约束条件。

【样例 6】

见下发文件 ex_jump6.in 和 ex_jump6.out，该测试用例满足子任务 3 的约束条件。

【样例 7】

见下发文件 ex_jump7.in 和 ex_jump7.out，该测试用例满足子任务 3 的约束条件。

【样例 8】

见下发文件 ex_jump8.in 和 ex_jump8.out。

【样例 9】

见下发文件 ex_jump9.in 和 ex_jump9.out。

【数据规模与约定】

全部数据均保证 $1 \leq n, q \leq 5 \times 10^5$, $1 \leq a_i, b_i \leq 10^9$, $1 \leq x \leq y \leq n + 1$ 。

本题采用子任务捆绑测试，各个子任务详细信息如下：

编号	子任务分数	$n, q \leq$	特殊性质
1	20	10	
2	10	5×10^5	对于所有 $i \in [1, n]$, 满足 $v_i = n - i + 1$
3	20	10^5	v_i 在所有可能的 $n!$ 种排列中均匀随机生成
4	50	5×10^5	

#3

传统	1000 ms	128 MiB
----	---------	---------

集了个合 (set)

【题目描述】

给定正整数 m , 求最大的 n , 使得存在 $S_1, S_2, \dots, S_m \subseteq \{1, 2, \dots, n\}$, 满足对于任意的 $j \in \{1, 2, \dots, n\}$ 都存在 $I \subseteq \{1, 2, \dots, m\}$ 使得 $\bigcap_{i \in I} S_i = \{j\}$ 。

请找出 n 的最大值, 构造对应的 S_1, S_2, \dots, S_m , 并为每个 j 给出对应的 I 。

【输入格式】

输入数据共一行, 包含一个整数 m 。

【输出格式】

先输出一行一个整数表示 n , 可以证明 $n \leq 2^m$ 。

接下来的 m 行依次表示 S_1, \dots, S_m 。其中第 i 行先输出一个整数表示 $|S_i|$, 再输出 $|S_i|$ 个互不相同的整数表示 S_i 中的元素。

接下来的 n 行的第 j 行中, 先输出一个正整数 k , 再输出 k 个整数 i_1, i_2, \dots, i_k , 它们需满足 $1 \leq i_1 < i_2 < \dots < i_k \leq m$ 且 $S_{i_1} \cap S_{i_2} \cap \dots \cap S_{i_k} = \{j\}$ 。

如果有多组合法的方案, 任意输出一组即可。

【样例数据】

【样例 1 输入】

```
1
```

【样例 1 输出】

```
1
1 1
1 1
```

【数据规模与约定】

本题共有 10 个测试点, 第 i 个测试点满足 $m = i$ 。每通过一个测试点得 10 分。

【提示】

可以使用下发文件中的 `chk.cpp` 以检验你的输出的合法性，使用时先将其编译为可执行文件 `chk`（在 Windows 系统下为编译为 `chk.exe`）。

- Linux 系统使用 `./chk <input-file> <output-file> <output-file>` 测试。
- Windows 系统使用 `chk <input-file> <output-file> <output-file>` 测试。

注意上述检验过程只会检验输出的合法性而并不会检验 n 是否是最大的，并且你需要确保 $1 \leq m \leq 10$ 和 $0 \leq n \leq 2^m$ 以保证该检验过程正常工作。

#4

传统	1000 ms	512 MiB
----	---------	---------

数树 (tree)

【题目描述】

有一棵包含 n 个顶点的树 $G = (V, E)$ 。

对于边的子集 S , 定义 forest S 为仅连接 S 中的边形成的森林, $f(S)$ 为 forest S 中每个连通块的重心的标号和。值得注意的是, 一个连通块可能有两个重心。

对于所有的 2^{n-1} 种 S , 请求出他们的 f 的和, 即 $\sum_{S \subseteq E} f(S)$ 的值。输出的答案对 998244353 取模。

【输入格式】

第一行输入一个整数 n , 表示树的大小。

接下来的 $n - 1$ 行每行一条输入两个整数 u 和 v , 表示点 u 和点 v 之间有一条连边。

【输出格式】

一行输出一个整数, 表示对 998244353 取模的答案。

【样例数据】

【样例 1 输入】

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

【样例 1 输出】

```
165
```

【样例 1 解释】

样例中输入的树结构如下:

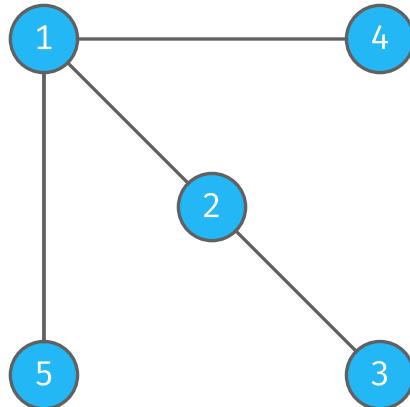


图 1: 样例 1 输入的树

【样例 2 输入】

```
10
1 3
2 5
6 7
2 6
3 6
4 6
9 5
5 8
1 10
```

【样例 2 输出】

```
18759
```

【样例 2 解释】

样例中输入的树结构如下：

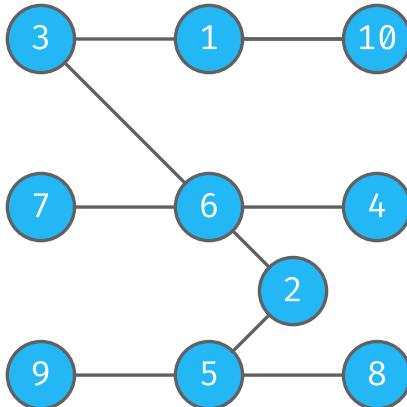


图 2: 样例 2 输入的树

【样例 3】

见下发文件 `ex_tree3.in` 和 `ex_tree3.out`, 该测试用例满足子任务 2 的约束条件。

【样例 4】

见下发文件 `ex_tree4.in` 和 `ex_tree4.out`, 该测试用例满足子任务 3 的约束条件。

【样例 5】

见下发文件 `ex_tree5.in` 和 `ex_tree5.out`, 该测试用例满足子任务 4 的约束条件。

【样例 6】

见下发文件 `ex_tree6.in` 和 `ex_tree6.out`。

【数据规模与约定】

全部数据均保证 $1 \leq n \leq 6000$, $1 \leq u, v \leq n$, 并且保证输入的边组成一颗树。

本题采用子任务捆绑测试, 各个子任务详细信息如下:

编号	子任务分数	$n, q \leq$
1	10	10
2	20	100
3	15	500
4	15	3000
5	40	6000