

2019 年全国青少年信息学奥林匹克 浙江省队选拔赛第一试

竞赛时间：3 月 27 日 8:00 – 13:00

题目名称	麻将	线段树	Minimax 搜索
目录	mahjong	segment	minimax
可执行文件名	mahjong	segment	minimax
输入文件名	mahjong.in	segment.in	minimax.in
输出文件名	mahjong.out	segment.out	minimax.out
每个测试点时限	2s	3s	2s
内存限制	512MB	512MB	512MB
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型
是否有附加文件	是	是	是

提交源程序必须加后缀

对于 C++ 语言	mahjong.cpp	segment.cpp	minimax.cpp
对于 C 语言	mahjong.c	segment.c	minimax.c
对于 Pascal 语言	mahjong.pas	segment.pas	minimax.pas

编译开关

对于 C++ 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 C 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 Pascal 语言	-O2	-O2	-O2

1 麻将

1.1 题目描述

九条可怜是一个热爱打麻将的女孩子。因此她出了一道和麻将相关的题目，希望这题不会让你对麻将的热爱消失殆尽。



今天，可怜想要打麻将，但是她的朋友们都去下自走棋了，因此可怜只能自己一个人打。可怜找了一套特殊的麻将，它有 $n(n \geq 5)$ 种不同的牌，大小分别为 1 到 n ，每种牌都有 4 张。

定义面子为三张大小相同或者大小相邻的麻将牌，即大小形如 $i, i, i (1 \leq i \leq n)$ 或者 $i, i+1, i+2 (1 \leq i \leq n-2)$ 。定义对子为两张大小相同的麻将牌，即大小形如 $i, i (1 \leq i \leq n)$ 。

定义一个麻将牌集合 S 是胡的当且仅当它的大小为 14 且满足下面两个条件中的至少一个：

- S 可以被划分成五个集合 S_1 至 S_5 。其中 S_1 为对子， S_2 至 S_5 为面子。
- S 可以被划分成七个集合 S_1 至 S_7 ，它们都是对子，且对应的大小**两两不同**。

举例来说，下列集合都是胡的（这儿只标记了大小）：

- $\{1, 1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9\}$
- $\{1, 1, 2, 2, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8\}$
- $\{1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7\}$

而下列集合都不是胡的：

- $\{1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9, 9\}$
- $\{1, 1, 1, 1, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8\}$
- $\{1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9, 11\}$

可怜先摸出了 13 张牌，并把剩下的 $4n - 13$ 张牌随机打乱。打乱是等概率随机的，即所有 $(4n - 13)!$ 种排列都等概率出现。

对于一个排列 P ，可怜定义 S_i 为可怜事先摸出的 13 张牌加上 P 中的前 i 张牌构成的集合，定义 P 的权值为**最小的 i 满足 S_i 存在一个子集是胡的**。如果你对麻将比较熟悉，不难发

现 P 的权值就是理论上的最早胡牌巡目数。注意到 $n \geq 5$ 的时候, S_{4n-13} 总是存在胡的子集的, 因此 P 的权值是良定义的。

现在可怜想要训练自己的牌效, 因此她希望你能先计算出 P 的权值的期望是多少。

1.2 输入格式

第一行输入一个整数 n , 表示这副特殊的麻将牌中的大小种类数。

接下来输入 13 行每行两个整数 $w, t (1 \leq w \leq n, 1 \leq t \leq 4)$, 表示可怜最开始摸出的第 i 张牌是大小为 w 的第 t 张牌, 保证 (w, t) 二元组两两不同。

1.3 输出格式

输出一行一个整数, 表示答案对 998244353 取模后的值。即如果答案的最简分数表示为 $\frac{x}{y} (x \geq 0, y \geq 1, \gcd(x, y) = 1)$, 你需要输出 $x \times y^{-1} \bmod 998244353$ 。

1.4 样例输入

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

1.5 样例输出

```
1
```

1.6 样例解释

上述牌型叫做纯正九莲宝灯, 不难发现不管再加一张什么牌它都是胡的。所以对于所有排列 P , 权值都是 1, 因此权值的期望就是 1。

1.7 数据范围与约定

对于 20% 的数据， $n = 5$ 。

对于 50% 的数据， $n \leq 13$ 。

对于另外 20% 的数据， $n \leq 100, w_i = i, t_i = 1$ 。

对于另外 20% 的数据， $n \leq 100, w_i = \lceil \frac{i}{4} \rceil, t_i = i \bmod 4 + 1$ 。

对于 100% 的数据， $5 \leq n \leq 100$ 。

2 线段树

2.1 题目描述

九条可怜是一个喜欢数据结构的女孩子，在常见的数据结构中，可怜最喜欢的就是线段树。

线段树的核心是懒标记，下面是一个带懒标记的线段树的伪代码，其中 tag 数组为懒标记：

```
1: function PUSHDOWN(Node)
2:   if tag[Node] = 1 then
3:     tag[Lson(Node)] ← 1
4:     tag[Rson(Node)] ← 1
5:     tag[Node] ← 0
6:   end if
7: end function
8:
9: function MODIFY(Node, l, r, ql, qr)
10:  if [l, r] ∩ [ql, qr] = ∅ then
11:    return
12:  end if
13:  if [l, r] ⊆ [ql, qr] then
14:    tag[Node] ← 1
15:    return
16:  end if
17:   $m \leftarrow \lfloor \frac{l+r}{2} \rfloor$ 
18:  PUSHDOWN(Node)
19:  MODIFY(Lson(Node), l, m, ql, qr)
20:  MODIFY(Rson(Node), m + 1, r, ql, qr)
21: end function
```

其中函数 Lson(Node) 表示 Node 的左儿子，Rson(Node) 表示 Node 的右儿子。

现在可怜手上有一棵 $[1, n]$ 上的线段树，编号为 1。这棵线段树上的所有节点的 tag 均为 0。接下来可怜进行了 m 次操作，操作有两种：

- 1 $l\ r$ ，假设可怜当前手上有 t 棵线段树，可怜会把每棵线段树复制两份（tag 数组也一起复制），原先编号为 i 的线段树复制得到的两棵编号为 $2i - 1$ 与 $2i$ ，在复制结束后，可怜手上一共有 $2t$ 棵线段树。接着，可怜会对所有编号为奇数的线段树进行一次 $\text{Modify}(\text{root}, 1, n, l, r)$ 。
- 2，可怜定义一棵线段树的权值为它上面有多少个节点 tag 为 1。可怜想要知道她手上所有线段树的权值和是多少。

2.2 输入格式

第一行输入两个整数 n, m 表示初始区间长度和操作个数。

接下来 m 行每行描述一个操作，输入保证 $1 \leq l \leq r \leq n$ 。

2.3 输出格式

对于每次询问，输出一行一个整数表示答案，答案可能很大，对 998244353 取模后输出即可。

2.4 样例输入

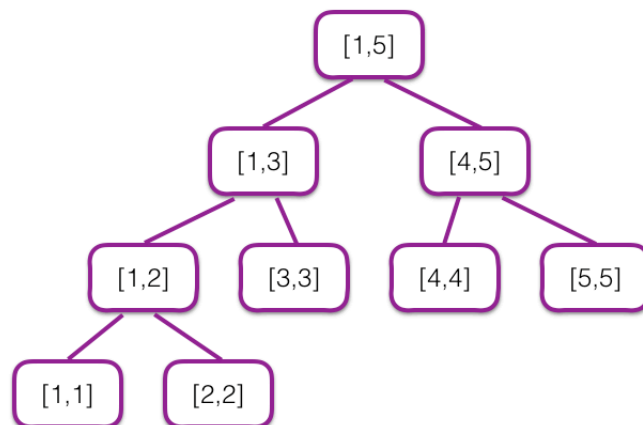
```
5 5
2
1 1 3
2
1 3 5
2
```

2.5 样例输出

```
0
1
6
```

2.6 样例解释

[1, 5] 上的线段树如下图所示：



在第一次询问时，可怜手上有一棵线段树，它所有点上都没有标记，因此答案为 0。

在第二次询问时，可怜手上有两棵线段树，按照编号，它们的标记情况为：

1. 点 [1, 3] 上有标记，权值为 1。
2. 没有点有标记，权值为 0。

因此答案为 1。

在第三次询问时，可怜手上有四棵线段树，按照编号，它们的标记情况为：

1. 点 $[1, 2], [3, 3], [4, 5]$ 上有标记，权值为 3。
2. 点 $[1, 3]$ 上有标记，权值为 1。
3. 点 $[3, 3], [4, 5]$ 上有标记，权值为 2。
4. 没有点有标记，权值为 0。

因此答案为 6。

2.7 数据范围与约定

测试点	n	m	其他约定
1	≤ 1000	≤ 10	无
2			
3		≤ 1000	
4			
5	$\leq 10^5$	$\leq 10^5$	询问只有一个
6			
7			
8			
9			无
10			

对于 100% 的数据， $1 \leq l \leq r \leq n$ 。

3 Minimax 搜索

3.1 题目描述

九条可怜是一个喜欢玩游戏的女孩子。为了增强自己的游戏水平，她想要用理论的武器武装自己。这道题和著名的 Minimax 搜索有关。

可怜有一棵有根树，根节点编号为 1。定义根节点的深度为 1，其他节点的深度为它的父亲的深度加一。同时在叶子节点权值给定的情况下，可怜用如下方式定义了每一个非节点的权值：

- 对于深度为奇数的非叶子节点，它的权值是它所有子节点的权值最大值。
- 对于深度为偶数的非叶子节点，它的权值是它所有子节点的权值最小值。

最开始，可怜令编号为 i 的叶子节点权值为 i ，并计算得到了根节点的权值为 W 。

现在，邪恶的 Cedyks 想要通过修改某些叶子节点的权值，来让根节点的权值发生改变。Cedyks 设计了一个量子攻击器，在攻击器发动后，Cedyks 会随机获得一个非空的叶子节点集合 S 的控制权，并可以花费一定的能量来修改 S 中的叶子节点的权值。

然而，修改叶子节点的权值也要消耗能量，对于 S 中的叶子节点 i ，它的初始权值为 i ，假设 Cedyks 把它的权值修改成了 w_i (w_i 可以是任意整数，包括负数)，则 Cedyks 在这次攻击中，需要花费的能量为 $\max_{i \in S} |i - w_i|$ 。

Cedyks 想要尽可能节约能量，于是他总是会**以最少的能量来完成攻击**，即在花费的能量最小的情况下，让根节点的权值发生改变。令 $w(S)$ 为 Cedyks 在获得了集合 S 的控制权后，会花费的能量。特殊地，对于某些集合 S ，可能无论如何改变 S 中叶子节点的权值，根节点的权值都不会发生改变，这时， $w(S)$ 的值被定义为 n 。为了方便，我们称 $w(S)$ 为 S 的稳定度。

当有 m 个叶子节点的时候，一共有 $2^m - 1$ 种不同的叶子节点的非空集合。在发动攻击前，Cedyks 想要先预估一下自己需要花费的能量。于是他给出了一个区间 $[L, R]$ ，他想要知道对于每一个 $k \in [L, R]$ ，有多少个集合 S 满足 $w(S) = k$ 。

3.2 输入格式

第一行输入三个整数 n, L, R ($n \geq 2, 1 \leq L \leq R \leq n$)。

接下来 $n - 1$ 行每行两个整数 u, v ，表示树上的一条边。

3.3 输出格式

输出一行 $R - L + 1$ 个整数，第 i 个整数表示 $w(S)$ 为 $L + i - 1$ 的集合 S 有多少个。答案可能会很大，请对 998244353 取模后输出。

3.4 样例输入

5 1 5

1 5
1 4
5 3
5 2

3.5 样例输出

4 0 1 0 2

3.6 样例解释

最开始，在可怜的设定下（ i 号叶子节点的权值为 i ），根节点的权值为 4。

树上一共有 3 个叶子节点 $\{2, 3, 4\}$ ，一共有 7 个非空的叶子节点权值，其中：

- $\{4\}, \{2, 4\}, \{3, 4\}, \{2, 3, 4\}$ 的稳定度为 1，只要稍微修改 4 号叶子节点的权值，根节点的权值就会发生改变。
- $\{2\}, \{3\}$ 的稳定度为 5，因为 5 号的权值是 2, 3 的较小值，在只修改 2 号或者 3 号的情况下，5 号点的权值始终小于等于 3，所以根节点的权值始终为 4。
- $\{2, 3\}$ 的稳定度为 3，要让根节点的权值发生改变，必须让 5 的权值大于 4，因此 w_2, w_3 都必须大于 4，所以稳定度为 3，一个可行的方案是把 w_2, w_3 都设为 5。

3.7 数据范围与约定

测试点	n	其他约定	测试点	n	其他约定
1	≤ 10	$L = R = n$	6	$\leq 2 \times 10^5$	$R - L \leq 50$
2	≤ 50		7		
3			8		无
4	≤ 5000		9		
5			10		

对于 100% 的数据，保证 $n \geq 2, 1 \leq L \leq R \leq n$ 。