

# NOIP 模拟赛

by neither\_nor

题目名称	midpoint displacement	erewrwerwer	缘
时间限制	1sec	1sec	1sec
内存限制	512MB	512MB	512MB
测试点数量	10	10	10

## midpoint displacement

题目描述:

**midpoint displacement** 算法是一种能够用于生成一维噪声的算法。

通俗地讲，你有一个长度为  $2^n+1$  的数组，初始时给你两端的权值，你要随机性地给每个下标赋一个权值，最终这个数组在统计学上会有一些优秀的性质。直观地讲，如果把权值视为高度的话，他会较为平滑而又有所起伏，仿佛层峦叠嶂在你眼前。

整个算法由若干次 **midpoint displacement** 不断进行来构成。

在一次 **midpoint displacement** 中，对于除了最后一个下标外的每一个已经被赋值的下标  $i$ ，找到他后面第一个已经被赋值的下标  $j$ ，将  $i$  和  $j$  的中点的权值设为  $i$  的权值与  $j$  的权值的平均数 + 一个随机数（可能为负，假设范围为  $[-lim, lim]$ ）

每进行一次 **midpoint displacement**，随机数的范围  $lim$  就会缩小一倍。

当所有下标都被赋值，算法结束。

为了测评方便，避免随机化与浮点数，在本题中使用光老犇随机器，一定会随机到最大的数，同时不取平均数，而是直接作和。也就是说，在 **midpoint displacement** 的时候，我们令中点的权值等于两端的权值的和加上  $lim$ ，模  $1e9+7$ 。并且，每进行完一次 **midpoint displacement**，我们将  $lim$  乘以 2，并模  $1e9+7$

在样例解释中你可以得到详细的解释。

现在给定  $n$ ，初始时两端的权值，以及随机数的初始范围，请计算最终每个点的权值和模  $1e9+7$

输入格式:

一行 4 个整数，分别代表  $n$ ，左端权值，右端权值，随机数的初始范围

输出格式:

一行一个整数，代表最终所有元素的和模  $1e9+7$

样例输入:

3 1 1 1

样例输出:

65

样例解释:

$n=3$ ，所以数组长度为  $2^n+1=9$

最终数组如下:

1 11 6 13 3 13 6 11 1

初始时左右两端为 1， $lim=1$

下标从 1 开始

$a[5]=a[1]+a[9]+lim=3$

$lim=lim*2=2$

$a[3]=a[1]+a[5]+lim=6$   
 $a[7]=a[5]+a[9]+lim=6$   
 $lim=lim*2=4$   
 $a[2]=a[1]+a[3]+lim=11$   
 $a[4]=a[3]+a[5]+lim=13$   
 $a[6]=a[5]+a[7]+lim=13$   
 $a[8]=a[7]+a[9]+lim=11$   
 所有点都被赋值，算法结束。  
 所有元素和为 65

提示： 请注意你的程序在进行连续加法时是否会超出 `int` 型整数的范围

数据范围与约定：

对于所有数据， $0 \leq \text{两端权值}$ 、 $lim \leq 32767$

测试点编号	n	其他约定
1	$\leq 23$	$a[1]=a[2^n+1]=lim=0$
2	$\leq 3$	
3	$\leq 3$	
4	$\leq 3$	
5	$\leq 16$	
6	$\leq 16$	
7	$\leq 16$	
8	$\leq 23$	
9	$\leq 23$	
10	$\leq 23$	

erewrwerwer

题目描述:



给你一个字符串，请问其有多少个子序列为 **erewrwerwer**

输入格式:

一行一个字符串，由小写字母'e'、'w'或者'r'组成

输出格式:

一行一个整数，表示为 **erewrwerwer** 的子序列数，模  $1e9+7$

样例输入:

**erewrwerwererewrwerwer**

样例输出:

**260**

数据范围与约定:

设  $n$  为字符串长度

对于前%20的数据,  $n \leq 11$

对于另外%30的数据,  $n \leq 20$

对于%100的数据,  $n \leq 100000$

# 缘

题目背景：



世界并不只如我们看到的一般，与这个世界共存着的，还有另外的一个世界与我们的世界相呼应。

在那幸福的瞬间，将会出现象征着美好的光之玉，如果能拿到的话，便能够实现一个愿望。

题目描述：

人与人之间的联系在两个世界都构成了一个以标号为 **1** 的人为根的有根树结构，这两个世界相互对应，在第一个世界标号为 **i** 的人，在第二个世界标号为 **p[i]**。而两个人之间的缘分就等于他们在两个世界的 **lca** 的深度和。

如果人与人之间有着很深的缘分，他们之间就会发生幸福的故事，这时天空之中就会出现象征着美好的光之玉，如果拿到就能够实现愿望。

现在你想知道，在这样的世界中，缘分最深的两个人的缘分有多少呢？

输入格式：

第一行一个整数 **n**，表示世界上一共有 **n** 个人

接下来一行 **n** 个整数，第 **i** 个代表 **p[i]**，保证 **p** 是 **1~n** 的排列

之后 **n-1** 行每行两个整数 **x[i]** 和 **y[i]**，表示标号为 **i** 的人在第一个世界里父亲是 **x[i]**，在第二个世界里父亲是 **y[i]** (**i** 从 **2** 到 **n**)

输出格式：

一行一个整数，表示缘分最深的两个人之间的缘分。

样例输入:

3

2 1 3

1 1

1 2

样例输出:

1

数据范围及约定:

对于 20%的数据,  $n \leq 100$

对于另外 30%的数据, 保证两个世界中树都是一条链

对于 100%的数据,  $n \leq 100000$