

2017 年全国青少年信息学奥林匹克

浙江省队选拔赛第二试

竞赛时间：4 月 28 日 8:00 – 13:00

题目名称	汉诺塔	线段树	字符串
目录	hanoi	segment	string
可执行文件名	hanoi	segment	string
输入文件名	hanoi.in	segment.in	string.in
输出文件名	hanoi.out	segment.out	string.out
每个测试点时限	3s	2s	3s
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	hanoi.cpp	segment.cpp	string.cpp
对于 C 语言	hanoi.c	segment.c	string.c
对于 Pascal 语言	hanoi.pas	segment.pas	string.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 汉诺塔

1.1 题目描述

虽然九条可怜经常出题，但是她发现要一个人出一场比赛是在是太难了，于是她丢下了手头的出题工作开始玩小游戏。

有一个游戏是汉诺塔的改版。在这个游戏中，盘子的大小可以相同，但是保证了相同大小的盘子恰好有 K 个。初始状态下，所有盘子自底向上从大到小地放置在第一根柱子上，对于相同大小的盘子，我们自底向上给它们从 1 到 K 标号。目标是把所有盘子都移动到第三根柱子上。

因为相同大小的盘子的上下顺序没有要求，不难发现合法的终止状态有很多，因此游戏里指定了一种终止状态。游戏的任务是最小化移动的步数。

可怜想了想觉得这个问题好像挺难的，于是她就把这个出到比赛里来给你做啦。

下面给出汉诺塔的具体规则：

- 有三根柱子和若干块圆盘，初始状态下所有圆盘都在第一根柱子上。
- 每一时刻你可以把某一根柱子顶端的圆盘移动到另一根柱子的顶端，目标是把所有盘子都移动到第三根柱子上。
- 移动的过程中必须要保证较大的圆盘不会压在较小的圆盘上。（相同大小的圆盘没有顺序限制）。

1.2 输入格式

第一行输入一个整数表示数据组数。

每组数据第一行是两个整数 n 和 K 。

接下来 n 行按照盘子从大到小的顺序给出终止状态下同一大小的盘子的相对顺序（从左到右依次表示自底向上），例如 $K = 2$ 时，2 1 表示原来在上方的盘子到了下面，原来在下方的盘子到了上面。

1.3 输出格式

对于每组数据输出一个整数表示最少步数。

1.4 样例输入

```
3
3 1
1
1
1
```

1 2
2 1
4 2
2 1
1 2
1 2
2 1

1.5 样例输出

7
2
31

1.6 数据范围与约定

测试点编号	K	测试点编号	K
1	= 1	6	= 3
2		7	
3	= 2	8	= 4
4		9	
5	= 3	10	

对于 100% 的数据，保证 $T \leq 10^4, 1 \leq n \leq 50, 1 \leq x_i \leq K$ 。

在实际的测试点中， T 组数据的 K 都是相同的。

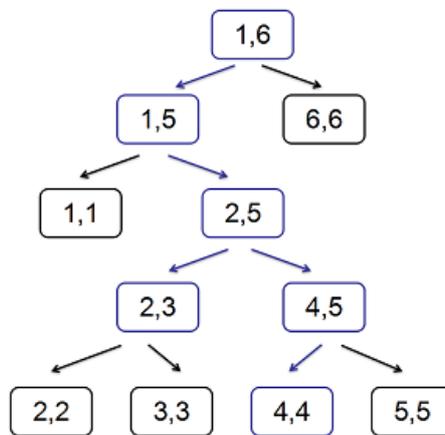
2 线段树

2.1 题目描述

线段树是九条可怜很喜欢的一个数据结构，它拥有着简单的结构、优秀的复杂度与强大的功能，因此可怜曾经花了很长时间研究线段树的一些性质。

最近可怜又开始研究起线段树来了，有所不同的是，她把目光放在了更广义的线段树上：在正常的线段树中，对于区间 $[l, r]$ ，我们会取 $m = \lfloor \frac{l+r}{2} \rfloor$ ，然后将这个区间分成 $[l, m]$ 和 $[m+1, r]$ 两个子区间。在广义的线段树中， m 不要求恰好等于区间的中点，但是 m 还是必须满足 $l \leq m < r$ 的。不难发现在广义的线段树中，树的深度可以达到 $O(n)$ 级别。

例如下面这棵树，就是一棵广义的线段树：



为了方便，我们按照先序遍历给线段树上所有的节点标号，例如在上图中， $[2, 3]$ 的标号是 5， $[4, 4]$ 的标号是 9，不难发现在 $[1, n]$ 上建立的广义线段树，它共有着 $2n - 1$ 个节点。

考虑把线段树上的定位区间操作（就是打懒标记的时候干的事情）移植到广义线段树上，可以发现在广义的线段树上还是可以用传统的线段树上的方法定位区间的，例如在上图中，蓝色节点和蓝色边就是在定位区间 $[2, 4]$ 时经过的点和边，最终定位到的点是 $[2, 3]$ 和 $[4, 4]$ 。

如果你对线段树不熟悉，这儿给出定位区间操作形式化的定义：给出区间 $[l, r]$ ，找出尽可能少的**区间互不相交**的线段树节点，使得它们区间的并集**恰好**是 $[l, r]$ 。

定义 $S_{[l,r]}$ 为定位区间 $[l, r]$ 得到的点集，例如在上图中， $S_{[2,4]} = \{5, 9\}$ 。定义线段树上两个点 u, v 的距离 $d(u, v)$ 为线段树上 u 到 v 最短路径上的边数，例如在上图中 $d(5, 9) = 3$ 。

现在可怜给了你一棵 $[1, n]$ 上的广义的线段树并给了 m 组询问，每组询问给出三个数 $u, l, r (l \leq r)$ ，可怜想要知道 $\sum_{v \in S_{[l,r]}} d(u, v)$ 。

2.2 输入格式

第一行输入一个整数 n 。

接下来一行包含 $n - 1$ 个空格隔开的整数：按照标号递增的顺序，给出广义线段树上所有**非叶子节点**的划分位置 m 。不难发现通过这些信息就能唯一确定一棵 $[1, n]$ 上的广义线段树。

接下来一行输入一个整数 m 。

之后 m 行每行输入三个整数 $u, l, r (1 \leq u \leq 2n - 1, 1 \leq l \leq r \leq n)$ ，表示一组询问。

2.3 输出格式

对于每组询问，输出一个整数表示答案。

2.4 样例输入

```
10
3 1 2 9 6 4 5 7 8
3
7 6 7
18 4 5
14 5 6
```

2.5 样例输出

```
7
11
3
```

2.6 数据范围与约定

测试点编号	n	m	其他约定
1	≤ 100	≤ 100	无
2	$\leq 2 \times 10^5$	≤ 20	
3		$\leq 2 \times 10^5$	$r = n$
4			
5			$u = 1$
6			
7			
8		无	
9			
10			

对于 100% 的数据，保证 $n \geq 2, m \geq 1$ 。

3 字符串

3.1 题目描述

猪小侠最近学习了字符串相关理论，现在他遇到了这样一个题：

维护一个动态字符串 $s[1..n]$ ，字符串的字符集是所有 $|x| \leq 10^9$ 的整数。要求支持两个操作：

- 1) 输入 l, r, d ，对于所有 $l \leq i \leq r$ ，将 $s[i]$ 修改为 $s[i] + d$ ，**注意 d 可能是负数**。
- 2) 输入 l, r ，输出子串 $s[l..r]$ 的**字典序最小的后缀**的起点位置。即，如果最小后缀是 $s[p..r]$ ，($l \leq p \leq r$)，请输出 p 。

3.2 输入格式

第一行两个非负整数 n, q 。

接下来一行包含 n 个正整数，表示初始时的字符串。

接下来 q 行，每行为 $1\ l\ r\ d$ 或 $2\ l\ r$ ，分别表示两种操作。

3.3 输出格式

对于所有的查询操作按顺序输出答案。

3.4 样例输入

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

3.5 样例输出

```
3
5
1
```

3.6 数据规模与约定

测试点编号	n	m	其他约定
1	≤ 300	≤ 300	无
2	$\leq 2 \times 10^4$	$\leq 10^4$	
3			
4	$\leq 2 \times 10^5$	$\leq 3 \times 10^4$	只有第二类操作
5			
6			数据随机生成
7			
8			无
9			
10			

对于 100% 的数据， $1 \leq l \leq r \leq n$ ， $|d| \leq 10^3$ ， $|s_i| \leq 10^8$ 。

注意，6 和 7 两个测试数据在随机生成时， s_i 在 $[0, 1]$ 中随机， d 在 ± 1 中随机。操作种类和操作区间都是等概率随机的。