

字符树 (tree)

【题目描述】

给你一个 n 个点的有根树，根为 1。每条边上有一个字符 $c = \{0, 1\}$ 。 S_u 表示从根到 u 的路径上每条边的字符依次写下来组成的字符串。保证每个节点向儿子的边上的字符互不相同。

对每个点 u ，有一个价值 val_u 和一个限制 a_u 。对每个点 u ，如果点 v 满足 S_u 是 S_v 的后缀。那么我们认为 v 是的 u 扩展点。

Alice 手里有一个字符串 S ，初始令 $S = S_u$ ，现在他可以删掉若干末尾的字符，使得 S 变成 S' 。并将 S' 告诉给 Bob。

Bob 获得了一个字符串 S' ，他需要在 S' 之后加入若干字符，并获得 S'' 。对于某个 u 的扩展点 v ，满足 $S'' = S_v$ ，并且 $|S'| \geq a_v$ ，那么 Bob 就获得了 val_v 的收益，当然 Bob 只能进行一次这样的操作，所以他会选择符合条件的 v 里， val_v 最大的那个。如果没有符合条件的 v ，Bob 只能获得 0 的收益。

现在 Alice 想知道，对于删除 $0 \sim |S|$ 个字符，总计 $|S| + 1$ 种删除方式里 Bob 能获得权值之和是多少？

对于每个 u ，你都需要回答 Alice 的询问。

形式化地说：

我们需要对每个点 u 求出 $ans_u = \sum_{0 \leq i \leq |S_u|} \max_{i \geq a_v \wedge S_u = S_v[|S_v| - |S_u| + 1, |S_v|] \wedge S_u[1, i] = S_v[1, i]} val_v$ 。

特殊的，如果对于某个 u ，不存在任何 v 满足条件，那么 $\max = 0$ 。

其中 $S[l, r]$ 表示字符串 S 的第 l 到第 r 个字符组成的字符串。特殊的， $S[x + 1, x]$ 表示空串。 $|S|$ 表示字符串 S 的长度， $\wedge$ 表示且。

【输入格式】

从文件 **tree.in** 中读入数据。

多组测试数据，第一行一个整数 T 表示数据组数。

对于每组测试数据，第一行一个正整数 n ，表示节点个数。

接下来 $n - 1$ 行，每行两个整数 fa_i, c_i 表示第 i 个点的父亲编号，以及边上的字符。

接下来一行 n 个正整数 $val_1, val_2, \dots, val_n$ 。

接下来一行 n 个非负整数 $a_1, a_2, \dots, a_n$ 。

【输出格式】

输出到文件 **tree.out** 中。

输出一行 n 个整数 $ans_1, ans_2, \dots, ans_n$ 。

【样例 1 输入】

```
1 1
2 5
3 1 0
4 1 1
5 2 0
6 2 1
7 1 2 3 4 5
8 0 1 0 1 2
```

【样例 1 输出】

```
1 3 4 6 8 5
```

【样例解释】

以下表格表示当 u, i 固定时，式子中 val_v 的最大值。

	$u = 1$	$u = 2$	$u = 3$	$u = 4$	$u = 5$
$i = 0$	3	0	3	0	0
$i = 1$	-	4	3	4	0
$i = 2$	-	-	-	4	5

【样例 2】

见下发文件中 *tree/tree2.in* 与 *tree/tree2.ans*。
这个样例满足测试点 3 ~ 5 的条件限制。

【样例 3】

见下发文件中 *tree/tree3.in* 与 *tree/tree3.ans*。
这个样例满足测试点 9 ~ 10 的条件限制。

【样例 4】

见下发文件中 *tree/tree4.in* 与 *tree/tree4.ans*。
这个样例满足测试点 11 ~ 12 的条件限制。

【数据范围】

对于所有数据保证 $1 \leq T \leq 5, 1 \leq n \leq 5 \times 10^5, 1 \leq val_i \leq 10^9, 1 \leq fa_i < i, c_i = \{0, 1\}, 0 \leq a_i \leq n$ 。

具体如下：

测试点编号	$n \leq$	特殊性质
1 ~ 2	100	无
3 ~ 5	2×10^3	
6 ~ 8	10^4	
9 ~ 10	10^5	A
11 ~ 12		B
13 ~ 16		无
17 ~ 20	5×10^5	

特殊性质 A： $c_i = 0$ 。

特殊性质 B： $fa_i = \lfloor \frac{i}{2} \rfloor$ 。