

Аллея

Input file: AVENUE.IN
Output file: AVENUE.OUT
exe-file: AVENUE.EXE

Time limit: 0.3 sec.
Memory limit: 8 M

В этой задаче мы будем сравнивать бинарные деревья. Бинарное дерево – это корневое дерево, в котором каждая вершина имеет не более двух потомков. Потомков какой-либо вершины x мы будем обозначать, как это принято, ЛевыйСын(x) и ПравыйСын(x). Бинарное дерево с корнем в вершине x будем обозначать $T(x)$.

Будем считать, что если у вершины x нет левого сына, то $T(\text{ЛевыйСын}(x))$ – пустое дерево, и, аналогично, если у вершины x нет правого сына, то $T(\text{ПравыйСын}(x))$ – пустое дерево.

Определим отношение “меньше” (которое будем обозначать знаком “ $<$ ”) для бинарных деревьев следующим образом:

1. Пустое дерево меньше любого непустого дерева.
2. Пусть $T(u)$ и $T(v)$ – два непустых бинарных дерева (разумеется, с корнями в вершинах u и v , соответственно). $T(u) < T(v)$ тогда и только тогда, когда выполняется одно из условий
 - 2.1. количество сыновей u меньше количества сыновей v
 - 2.2. v и u имеют одинаковое количество сыновей, и при этом
 - 2.2.1 $T(\text{ЛевыйСын}(u)) < T(\text{ЛевыйСын}(v))$, либо
 - 2.2.2 $T(\text{ЛевыйСын}(u)) = T(\text{ЛевыйСын}(v))$, а $T(\text{ПравыйСын}(u)) < T(\text{ПравыйСын}(v))$.

Задана последовательность из N ($N \leq 25000$) непустых бинарных деревьев, каждое из которых имеет не более 30 вершин. Требуется найти в ней возрастающую подпоследовательность наибольшей возможной длины.

Более формально: задана последовательность непустых бинарных деревьев T_1, T_2, \dots, T_N , каждое дерево имеет не более 30 вершин. Требуется найти такую последовательность $1 \leq k_1 < k_2 < \dots < k_p \leq N$, что $T_{k_1} < T_{k_2} < \dots < T_{k_p}$, при этом p должно быть максимально возможным.

Входные данные. В первой строке входного файла расположено одно целое положительное число N – количество деревьев в ряду, $1 \leq N \leq 25000$. Каждая из следующих N строк содержит описание одного двоичного дерева. Для этого вершины в каждом дереве пронумерованы натуральными числами от 1 до M , где M – количество вершин в дереве. Описание дерева представляет из себя ряд из $2M$ целых чисел из диапазона от 0 до M : сначала идёт номер вершины ЛевыйСын(1), затем вершины ПравыйСын(1), затем идут номера левого и правого сыновей вершины номер 2, затем вершины номер 3 и т.д. Если у вершины нет того или иного сына, то на соответствующем месте стоит 0. Соседние числа в описании дерева разделены одним пробелом.

Выходные данные. В первой строке выходного текстового файла должно находиться одно целое положительное число P – наибольшая возможная длина возрастающей последовательности бинарных деревьев. Внимание! Эта часть задачи оценивается отдельно от следующей.

В следующих P строках следует выводить значения k_1, k_2, \dots, k_p по одному числу в строке – номера деревьев, образующих самую длинную возрастающую подпоследовательность. Если задача имеет несколько решений, то следует выводить любое одно из них.

Примеры

AVENUE.IN

```
5
0 0 0 0 1 2 3 0
4 0 3 1 0 0 0 0
0 0 4 0 0 1 3 0
0 0 0 0 1 4 2 0
0 4 0 0 0 0 3 2
```

AVENUE.IN

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

AVENUE.OUT

```
2
1
4
```

AVENUE.OUT

```
3
3
4
6
```

Примечания.

1. Другие примеры возрастающей подпоследовательности: 1, 2 или 3, 4
2. Заметим, что второе и четвёртое деревья равны между собой.