max planck institut
informatik

Competitive Programming

Saarland University — Summer Semester 2020

Julian Baldus, Markus Bläser, Karl Bringmann, Marian Dietz, Simon Schwarz,
Christoph Weidenbach, Dominic Zimmer

Assignments Week 2

Deadline: **May 19, 2020 at 16:00 sharp**

Please submit solutions to the problems in our judge system, available at
`https://compro.mpi-inf.mpg.de/`.
You can find your credentials on your personal status page in our CMS.

Problem	`sumup`	`multiset`	`covering`	`simplify`
Points	3	3	3	3
Difficulty				
Time Limit	1s	8s	30s	20s
Memory Limit	2 GB	4 GB	2 GB	4 GB

Please note:

- *Later, we will reopen the judge for the problems of last week. However, you won't get any points for submissions of these problems.*
- *In the judge you can switch between the exercises of different weeks in the top-right corner.*
- Your solution will be judged immediately after submitting. This may take some time, depending on the current server load.
- You can submit as many times as you want. However, don't abuse the server or try to extract the secret test cases.
- If your solution is **accepted**, you will receive the points specified in the table above.
- If you get **another verdict**, you will receive 0 points.

Tree Calculations: Summing-up

Problem ID: sumup

This problem needs a parser that can be used for the other problems on trees.

Given is a collection of trees where inner nodes of trees are labelled with $*$ or $+$ and leafs are labelled with natural numbers. Every inner node has at least two children.

Calculate the value of a tree by multiplying all values of subtrees of a $*$ node, by summing-up all values of subtrees of a $+$ node, and considering the natural numbers as values for the leafs.

Output the overall sum of all single tree values.

Input

A tree is given by the grammar

$$
\begin{array}{rcl}
\langle \text{tree} \rangle & := & \langle \text{andor} \rangle \text{ “(” } \langle \text{stree} \rangle \text{ “,” } \langle \text{stree} \rangle \text{ } \{ \text{ “,” } \langle \text{stree} \rangle \text{ } \}^* \text{ “)”} \\
\langle \text{stree} \rangle & := & \langle \text{nat} \rangle \mid \langle \text{tree} \rangle \\
\langle \text{andor} \rangle & := & \text{“}*\text{”} \mid \text{“+”} \\
\langle \text{nat} \rangle & := & \text{natural number starting from 1}
\end{array}
$$

The input is a collection of trees, one per line. After the last tree, there is another newline before the end of file.

Output

The sum of all tree values.

Constraints

- $1 \leq n < 100$ for the values n at the leafs.
- The input consists of at most 1000 trees.
- The depth of a tree is at most 4.
- A parent node has at most 5 children.
- The sum of all tree values is less than 10^{18}.

Sample Input 1	Sample Output 1
`+(4,*(2,3))`	`10`

Sample Input 2	Sample Output 2
`*(3,+(4,+(3,7,6)))` `+(5,*(2,+(7,6,2),1))`	`95`

Tree Calculations: Multi-Set Comparison

Problem ID: multiset

This problem can share the parser with problem Summing-Up.

Assume two collections of trees where leafs are labelled with natural numbers. Check whether the two collections are identical if considered as multi-sets of trees.

Two trees are considered equal if and only if they contain equal subtrees in the same order. Two leafs are equal if they share the same value.

Input

A tree is given by the grammar

$$
\begin{aligned}
\langle \text{tree} \rangle &:= \langle \text{andor} \rangle \text{ “(” } \langle \text{stree} \rangle \text{ “,” } \langle \text{stree} \rangle \{ \text{ “,” } \langle \text{stree} \rangle \}^* \text{ “)”} \\
\langle \text{stree} \rangle &:= \langle \text{nat} \rangle \mid \langle \text{tree} \rangle \\
\langle \text{andor} \rangle &:= \text{“}\star\text{”} \mid \text{“+”} \\
\langle \text{nat} \rangle &:= \text{natural number starting from 1}
\end{aligned}
$$

The input consists of two collections of trees. The two collections are separated by a single line consisting only of “0”. Each collection contains a list of trees, one per line.

Output

Either “`Equal`” or “`UnEqual`”.

Constraints

- $1 \leq n \leq 500$ for the values n at the leafs.
- The overall number of trees in the input does not exceed 200000.
- The total number of nodes in all input trees (i.e. the number of occurrences of “tree” and “nat”) does not exceed $25 \cdot 10^6$.

You can find two additional sample files for download in the online judge. All large secret test cases are variations of the provided cases, i.e. if your program is fast enough on the provided cases, you can assume that it will be fast enough for all cases. For this task, we will not try to test your worst case running time.

Please download the files one by one instead of downloading the zip with all samples.

Sample Input 1	Sample Output 1

`+(3,*(4,113,7))`
`*(50,500)`
`+(7,5)`
`+(7,5)`
`0`
`+(7,5)`
`*(50,500)`
`+(3,*(4,113,7))`
`+(7,5)`
	`Equal`

Sample Input 2

```
+(3,*(4,113,7))
*(50,500)
+(7,5)
+(7,5)
0
*(50,500)
+(3,*(4,113,7))
+(7,5)
```

Sample Output 2

```
UnEqual
```

Sample Input 3

```
+(3,*(4,113,7))
*(50,500)
+(7,5)
+(7,5)
0
*(7,5)
*(50,500)
+(3,*(4,113,7))
+(7,5)
```

Sample Output 3

```
UnEqual
```

Sample Input 4

```
+(3,*(4,113,7))
*(50,500)
+(7,5)
+(7,5)
0
*(50,500)
+(7,5)
+(4,*(3,113,7))
+(7,5)
```

Sample Output 4

```
UnEqual
```

Tree Calculations: Domination

Problem ID: covering

This problem can share the parser with problem Summing-Up and problem Multi-Set Comparison.

Let set(s) be the set of all natural numbers of the tree s. For example, set$(+(6, *(*(6, 3), 4))) = \{6, 3, 4\}$. We say that a tree s_1 *dominates* a tree s_2 if set$(s_2) \subseteq$ set(s_1).

You are given a collection of trees. Under all of them, find the tree that dominates the maximal number of trees from the collection excluding itself. Output the number of trees it dominates.

Input

A tree is given by the grammar

$$\begin{aligned} \langle \text{tree} \rangle &:= \langle \text{andor} \rangle \text{ “(” } \langle \text{stree} \rangle \text{ “,” } \langle \text{stree} \rangle \text{ } \{ \text{ “,” } \langle \text{stree} \rangle \text{ } \}^* \text{ “)”} \\ \langle \text{stree} \rangle &:= \langle \text{nat} \rangle \mid \langle \text{tree} \rangle \\ \langle \text{andor} \rangle &:= \text{“}*\text{”} \mid \text{“+”} \\ \langle \text{nat} \rangle &:= \text{natural number starting from 1} \end{aligned}$$

The input is a collection of trees, one per line. After the last tree, there is another newline before the end of file.

Output

A natural number, indicating the maximum number of trees from the collection dominated by one of them.

Constraints

- $1 \leq n \leq 500$ for the values n at the leafs.
- The overall number of trees in the input does not exceed 10000.
- The total number of nodes in all input trees (i.e. the number of occurrences of “tree” and “nat”) does not exceed $6.5 \cdot 10^6$.

You can find two additional sample files for download in the online judge. All large secret test cases are variations of the provided cases, i.e. if your program is fast enough on the provided cases, you can assume that it will be fast enough for all cases. For this task, we will not try to test your worst case running time.

Explanation for the sample below: The second tree dominates the third and fourth tree. All other trees do not dominate any tree.

Sample Input 1	Sample Output 1

`*(50,500)`
`+(3,*(4,113,7))`
`*(7,4)`
`+(7,3)`
	`2`

Tree Calculations: Simplification

Problem ID: simplify

You can reuse the parser of problems Summing-Up, Multi-Set Comparison, and Domination.
Assume an and/or tree where inner nodes are labelled with $*$ or $+$ and leafs are labelled with natural numbers, this time including 0. Every inner node has at least two children. The number of children is not a priori limited.

The trees represent so called *and/or* trees where $*$ stands for logical *and* and $+$ stands for logical *or*. The number 1 represents *true* and the number 0 represents *false* and all other numbers represent propositional variables.

Now consider the following simplification rules on an and/or tree where s, s_i stand for arbitrary trees and the rules apply to any node of the tree in any order:

$$
\begin{aligned}
*(1, s) &\rightarrow s \\
+(0, s) &\rightarrow s \\
*(s, 1) &\rightarrow s \\
+(s, 0) &\rightarrow s \\
*(s_1, \ldots, s_k, 1, s_{k+1}, \ldots, s_n) &\rightarrow *(s_1, \ldots, s_k, s_{k+1}, \ldots, s_n) \quad \text{for } n > 1 \\
*(s_1, \ldots, s_k, 0, s_{k+1}, \ldots, s_n) &\rightarrow 0 \\
+(s_1, \ldots, s_k, 0, s_{k+1}, \ldots, s_n) &\rightarrow +(s_1, \ldots, s_k, s_{k+1}, \ldots, s_n) \quad \text{for } n > 1 \\
+(s_1, \ldots, s_k, 1, s_{k+1}, \ldots, s_n) &\rightarrow 1
\end{aligned}
$$

Input

A tree is given by the grammar

$$
\begin{aligned}
\langle \text{tree} \rangle &:= \langle \text{andor} \rangle \text{ “(” } \langle \text{stree} \rangle \text{ “,” } \langle \text{stree} \rangle \text{ \{ “,” } \langle \text{stree} \rangle \text{ \}* “)” } \\
\langle \text{stree} \rangle &:= \langle \text{nat} \rangle \mid \langle \text{tree} \rangle \\
\langle \text{andor} \rangle &:= \text{“*” } \mid \text{ “+”} \\
\langle \text{nat} \rangle &:= \text{natural number starting from 0}
\end{aligned}
$$

The input consists of exactly one tree in one line. It follows a newline before the end of file.

Output

The tree after exhaustive application of the simplification rules to all nodes of the tree. Please note that all rules preserve the argument (children) order of $+$ and $*$ nodes. Your solution should preserve the order as well.

Constraints

- $0 \leq n \leq 500$ for the values n at the leafs.
- The number of nodes in the input tree (i.e. the number of occurrences of “tree” and “nat”) does not exceed $25 \cdot 10^6$.

Bonus no extra points

If you use C++, make sure your version runs with `valgrind` without memory leaks. Delete trees that are not needed anymore on the fly and remember to also delete the top-level tree.

Sample Input 1	Sample Output 1
`*(1,+(2,3,0,4),*(3,4),*(+(1,4),5,1,2),*(1,9))`	`*(+(2,3,4),*(3,4),*(5,2),9)`

Sample Input 2	Sample Output 2
`+(*(+(3,4,1),2),*(2,+(0,2)))`	`+(2,*(2,2))`