439. Ternary Expression Parser

Medium 231 30 Add to List Share

Given a string representing arbitrarily nested ternary expressions, calculate the result of the expression. You can always assume that the given expression is valid and only consists of digits `0-9`, `?`, `:`, `T` and `F` (`T` and `F` represent True and False respectively).

Note:

1. The length of the given string is ≤ 10000.
2. Each number will contain only one digit.
3. The conditional expressions group right-to-left (as usual in most languages).
4. The condition will always be either `T` or `F`. That is, the condition will never be a digit.
5. The result of the expression will always evaluate to either a digit `0-9`, `T` or `F`.

Example 1:

```
Input: "T?2:3"

Output: "2"

Explanation: If true, then result is 2; otherwise result is 3.
```

Example 2:

```
Input: "F?1:T?4:5"

Output: "4"

Explanation: The conditional expressions group right-to-left. Using parenthesis, it is read/evaluated as:

            "(F ? 1 : (T ? 4 : 5))"                 "(F ? 1 : (T ? 4 : 5))"
         -> "(F ? 1 : 4)"                 or     -> "(T ? 4 : 5)"
         -> "4"                                  -> "4"
```

Example 3:

```
Input: "T?T?F:5:3"

Output: "F"

Explanation: The conditional expressions group right-to-left. Using parenthesis, it is read/evaluated as:

            "(T ? (T ? F : 5) : 3)"                 "(T ? (T ? F : 5) : 3)"
         -> "(T ? F : 3)"                 or     -> "(T ? F : 5)"
         -> "F"                                  -> "F"
```

Accepted 18,549 | Submissions 33,464

Seen this question in a real interview before? Yes No

Contributor

Companies

Related Topics

Similar Questions

```
class Solution(object):
    def parseTernary(self, expression):
        """
        :type expression: str
        :rtype: str
        """

```