

☑ PAT 乙级题目讲解：1003 《我要通过！》

🧩 题目简介

本题属于字符串合法性判断问题，题目要求根据字符串的结构判断其是否为一个“正确”的字符串。所谓“正确”，是指字符串满足某种特定的格式，由字符 **P**、**A**、**T** 组成，且要满足一系列生成规则。

我们需要判断每一个字符串是否满足如下条件：

1. 只能包含字符 **P**、**A**、**T**；
2. 且仅出现一次 **P** 和一次 **T**，且 **P** 必须出现在 **T** 之前；
3. 将字符串表示为 **aPbTc** 的形式，且中间部分（即 **b**）长度至少为 1，并满足公式： $\text{len}(a) * \text{len}(b) == \text{len}(c)$ 。

🔧 样例分析

输入：

```
10
PAT
PAAT
AAPATAA
AAPAATAAAA
xPATx
PT
Whatever
APAAATAA
APT
PTAA
```

逐个分析如下：

- **PAT**： $\text{len}(a)=0, \text{len}(b)=1, \text{len}(c)=0 \Rightarrow 0 \times 1=0$ ☑
- **PAAT**： $\text{len}(a)=0, \text{len}(b)=2, \text{len}(c)=0 \Rightarrow 0 \times 2=0$ ☑
- **AAPATAA**： $\text{len}(a)=2, \text{len}(b)=1, \text{len}(c)=2 \Rightarrow 2 \times 1=2$ ☑
- **AAPAATAAAA**： $\text{len}(a)=2, \text{len}(b)=2, \text{len}(c)=4 \Rightarrow 2 \times 2=4$ ☑
- **xPATx**：含非法字符 ✕
- **PT**：中间无 **A** $\Rightarrow \text{len}(b)=0$ ✕
- **Whatever**：含非法字符 ✕
- **APAAATAA**： $\text{len}(a)=1, \text{len}(b)=3, \text{len}(c)=2 \Rightarrow 1 \times 3 \neq 2$ ✕
- **APT**：中间无 **A** ✕
- **PTAA**：多个 **T** ✕

因此输出为：

YES
YES
YES
YES
NO
NO
NO
NO
NO
NO

数学推导：为何 $\text{lena} * \text{lenb} == \text{lenc}$

我们设一个基础合法串为 `PAT`，其中 $\text{len}(a)=0$ ， $\text{len}(b)=1$ ， $\text{len}(c)=0$ ，此时 $0 \times 1 = 0$ 成立。

考虑生成新串的过程，每次在中段 `b` 末尾增加一个 `A`，同时在尾部 `c` 增加一段等长的 `A`。我们来推导该规则的数学本质：

设：

- 初始时： $\text{len}(a) = na$ ， $\text{len}(b) = nb = 1$ ， $\text{len}(c) = nc = na$
- 每次扩展中段 `b` 增加一个 `A`，变成 $nb + m$
- 同时末尾 `c` 要扩展为： $nc + m \times na$ （每次扩展 `b`，就多加一段 `a` 的长度）

于是：

最终的中段长度： $\text{len}(b) = nb + m$

最终的尾段长度： $\text{len}(c) = nc + m \times na = na + m \times na = na \times (1 + m)$

此时：

$$\begin{aligned}\text{len}(b) &= 1 + m \\ \text{len}(c) &= na \times (1 + m)\end{aligned}$$

所以有：

$$\text{len}(a) \times \text{len}(b) = na \times (1 + m) = \text{len}(c)$$

因此，所有由基本串扩展得到的串都满足：

$$\text{len}(a) \times \text{len}(b) == \text{len}(c)$$

这个公式正是我们在程序中进行判断的关键逻辑。

解题思路

变量说明

变量名	含义
n	测试用例数量
s	当前待检测的字符串
p	字符 P 所在的位置
t	字符 T 所在的位置
cp	字符 P 出现次数
ct	字符 T 出现次数
lena	P 左侧 A 的个数
lenb	P 和 T 之间 A 的个数
lenc	T 右侧 A 的个数

☑ Step 1: 过滤非法字符

遍历字符串中所有字符，若出现非 P、A、T 以外的字符，立即判为非法。

```
for(int i = 0; i < len; i++){
    if(s[i] != 'P' && s[i] != 'A' && s[i] != 'T'){
        f = 0;
        break;
    }
}
```

☑ Step 2: 统计 P、T 出现次数及位置

确保 P 和 T 各自出现且仅出现一次，且 P 必须在 T 之前。

```
if(cp != 1 || ct != 1 || p >= t) f = 0;
```

☑ Step 3: 根据公式判断是否合法

将字符串分段后，统计三部分 A 的数量，判断是否满足 $lena * lenb == lenc$ 且 $lenb > 0$ 。

```
int lena = p;
int lenb = t - p - 1;
int lenc = len - t - 1;
if(!lenb || lena * lenb != lenc) f = 0;
```

完整代码

```
#include<bits/stdc++.h>
using namespace std;

int n;
string s;
```

```

int main(){
    cin >> n;
    while(n--){
        cin >> s;
        int len = s.size();
        bool f = 1;
        // 1. 只有 P A T
        int p, t, cp = 0, ct = 0;
        for(int i = 0; i < len; i++){
            if(s[i] != 'P' && s[i] != 'A' && s[i] != 'T'){
                f = 0;
                break;
            }
            if(s[i] == 'P'){
                cp++;
                p = i;
            }
            if(s[i] == 'T'){
                ct++;
                t = i;
            }
        }
        // 2. P 和 T 唯一, 且 P 先于 T
        if(cp != 1 || ct != 1 || p >= t) f = 0;
        // 3. aPbTc -> lenb >= 1 且 lena * lenb = lenc
        int lena = p;
        int lenb = t - p - 1;
        int lenc = len - t - 1;
        if(!lenb || lena * lenb != lenc) f = 0;
        if(f) cout << "YES";
        else cout << "NO";
        if(n > 0) cout << "\n";
    }
    return 0;
}

```

🚫 常见错误提醒

错误类型	具体表现
字符合法性错误	包含 P、A、T 以外的字符
P、T 多次出现	cp != 1 或 ct != 1
P 在 T 之后	p >= t
中间部分无 A	lenb == 0
数学公式不成立	lena * lenb != lenc
输出格式错误	多输出换行或少输出换行

✅ 总结归纳

- 本题是构造字符串合法性判断的经典题型。
- 推导出公式 $\text{lena} * \text{lenb} == \text{lenc}$ 是解题核心。
- 熟练处理字符串分段，逻辑判断，数学建模。

思维拓展

- 本题可延展到“语言生成规则”建模、“有限状态机”合法串识别问题。
- 可作为学习**自底向上归纳规则**的入门训练题。