

C1-A problem

题目描述

小水獭正在学习分治和递归，在计算复杂度时用到了主定理，希望你可以帮它写一个程序来计算满足特定形式递归式的算法复杂度。

具体来说，给定正整数 a, b, k ，求递归式：

$$T(n) = aT(n/b) + O(n^k)$$

根据主定理，有：

$$T(n) = \begin{cases} O(n^{\log_b a}), & \log_b a > k \\ O(n^k \log n), & \log_b a = k \\ O(n^k), & \log_b a < k \end{cases}$$

题解思路

- 首先依据题意，此题只有三种输出答案，将他们分别存储为字符串。
- 根据输入的 a, b, k 三个正整数判断应输出的字符串即可。【注意由于对数函数的精度问题，需要将 $\log_b(a) = k$ 转换为 $a = b^k$ ，其他两种判断类似】

代码

```
#include <iostream>
#include <math.h>
using namespace std;

int main() {
    int n;
    string str1 = "n^{\\log_b a}";
    string str2 = "n^k \\log n";
    string str3 = "n^k";
    cin >> n;
    for (int i = 0; i < n; i++) {
        int a, b, k;
        cin >> a;
        cin >> b;
        cin >> k;
        if (a < pow(b, k)) {
            cout << str3 << endl;
        }
        else if (a == pow(b, k)) {
            cout << str2 << endl;
        }
    }
}
```

```
        else {  
            cout << str1 << endl;  
        }  
    }  
    return 0;  
}
```