矩阵快速幂的 JAVA 实现

作者：chillax
原文链接：https://ld246.com/article/1484566255859
来源网站：链滴

矩阵快速幂

题目描述：

给定一个n*n的矩阵，求该矩阵的k次幂，即P^k。

**如何快速的算出一个矩阵的N次幂呢，举个例子，比如A^19 => （A^16）* A^2）*（A^1）显然采取这样的方式计算时因子数将是log(n)级别的(原来的因子数是n)，不这样，因子间也是存在某种联系的。

比如A^4能通过(A^2)*(A^2)得到，A^8又能通过(A^4)* A^4)得到，这点也充分利用了现有的结果作为有利条件。

下面举个例子进行说明：现在要求A^156,而156(10)=10011100(2) 也就有A^156=>(A^4)* A^8)*(A^16)*(A^128) 考虑到因子间的联系，我们从二进制10011100中 最右端开始计算到最左端。**

以下代码就是其实现方式：

```
public class MatrixMulti {

    public static long[][] mut(int k,int n,long[][] A){
        long [][] res = new long[n][n];
        for(int i = 0 ; i < res.length ;i++){
            for(int j = 0 ; j< res[i].length ;j++){
                if(i==j){
                    res[i][j] = 1;
                }else{
                    res[i][j] = 0;
                }
            }
        }
        while(k!=0){
            if((k&1)==1) res = f(res,A);
            k>>=1;
            A = f(A,A);
        }
        return res;
    }

    public static long[][] f(long[][] A,long[][] B){
        long res[][] = new long[A.length][B.length];
        for(int i = 0 ; i < res.length ;i++){
            for(int j = 0 ; j< res[i].length ;j++){
                for(int k = 0 ; k < A[0].length ;k++){
                    res[i][j] += A[i][k]*B[k][j];
                }
            }
        }
        return res;
    }

}
```

其中函数f(),是定义了矩阵的乘法运算，在矩阵中，单位矩阵就是相当于常数1，在函数mut（）中，主要的就是while循环里的代码，运用了分治的思想，快速的计算矩阵的n次幂。好吧，矩阵快速幂的解就到这里吧。