剑指Offer\_16\_数值的整数次方

# 剑指Offer\_16\_数值的整数次方

## 题目介绍

\* 题目介绍：

\* 给定一个double类型的浮点数base和int类型的整数exponent。求base的exponent次方。

## 思路分析

考虑特殊情况：0的0次方没有意义。

指数是负数、0、正数分别处理。

例如求2的32次方，普通需要连乘31次，优化依次求2的2次方、4次方、8次方、16次方、32次方。利用递归即可实现。

优化：乘以2或除以2利用左移和右移实现；

对2求余利用与实现，如((expoent&1) == 1)。

## Java代码

/\*\*

\* 剑指Offer16：数值的整数次方

\*/

@Test

public void test(){

double base = 2;

int exponent = 4;

double result = Power(base,exponent);

System.out.println("result = " + result);

}

boolean isFalse = false;//结果是否错的

public double Power(double base, int exponent) {

if(base==0&&exponent==0) {//0的零次方没有意义，默认返回0

isFalse = true;//返回0的不一定只是0的0次方，所以需要判断结果为0是否是错的；更严谨

return 0;

}

if(exponent==0) return 1;

double result = 0;

if(exponent < 0){

result = myPower(base,-exponent);

result = 1/result;

}else{

result = myPower(base,exponent);

}

return result;

}

/\*\*

\* 利用递归实现乘方

\*/

public double myPower(double base,int expoent){

if(expoent == 1) return base;

if(expoent == 0) return 1;

double result = myPower(base,expoent>>1);

//(expoent&1) == 1 利用位运算实现exponent%2的功能，效率更高

return ((expoent&1) == 1)? result\*result\*base:result\*result;

}

/\*\*

\* 普通的求乘方函数

\*/

/\* public double myPower2(double base,int expoent){

double result = 1;

while(expoent-- != 0){

result \*= base;

}

return result;

}\*/