第4章 原根与指数参考答案

计算证明

1. 求2^{12}对模37的次数.

解　$\varphi(37)=36=2^2\times 3^2$，则36的因子有 1,2,3,4,6,9,12,18,36. 依次求得：

$2^1\equiv 2\pmod{37}$　$2^2\equiv 4\pmod{37}$　$2^4\equiv 16\pmod{37}$　$2^6\equiv 27\pmod{37}$

$2^9\equiv 31\pmod{37}$　$2^{12}\equiv 26\pmod{37}$　$2^{18}\equiv 36\pmod{37}$ 则有：$\mathrm{ord}_{37}(2)=\varphi(37)=36$.

故$\mathrm{ord}_{37}(2^{11})=\dfrac{\mathrm{ord}_{37}(2)}{(\mathrm{ord}_{37}(2),12)}=3$.

2. 求模61的最小非负完系中所有次数为4的整数.

解　设$x(x\in\mathbb{Z},0<x<61)$满足$\mathrm{ord}_{61}(x)=4$，即有$x^4\equiv 1\pmod{61}$且$x^j\not\equiv 1\pmod{61}(j=1,2,3)$.

得到$(x^2-1)(x^2+1)\equiv 0\pmod{61}$，而$x^2\not\equiv 1\pmod{61}$，必有$x^2+1\equiv 0\pmod{61}$.

解得$x\equiv 11,50\pmod{61}$（即$x^2\equiv 121\pmod{61}$）.

经验证，当x取11,50时，$x^j\not\equiv 1\pmod{61}(j=1,2,3)$均满足 . (**必须要验证，否则不能保证“最小”**)

故11,50即为所求.

3. 设$ab\equiv 1\pmod{m}$，求证：$\mathrm{ord}_m(a)=\mathrm{ord}_m(b)$.

证明　（由a,b的对称性可知结论显然成立. 证毕）（**不给分**）

由$ab\equiv 1\pmod{m}$得$b\equiv a^{-1}\pmod{m}$，则$b^{\mathrm{ord}_m(a)}=a^{-\mathrm{ord}_m(a)}=(a^{\mathrm{ord}_m(a)})^{-1}\equiv 1\pmod{m}$.

下面用反证法证明$\mathrm{ord}_m(a)$也是b的次数.

假设　$\mathrm{ord}_m(b)=r<\mathrm{ord}_m(a)$　$(r>0)$，由　$ab\equiv 1\pmod{m}$　得　$a\equiv b^{-1}\pmod{m}$，则$a^r=b^{-r}=(b^r)^{-1}\equiv 1\pmod{m}$. 说明$r$是$a$的次数. 矛盾，假设不成立. 证毕.

4. 设　a,b,m是正整数，如果　a,b分别与　m互素，且满足　$(\mathrm{ord}_m(a),\mathrm{ord}_m(b))=1$，证明$\mathrm{ord}_m(ab)=\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)$.

证明　由题易知：$\begin{cases}a^{\mathrm{ord}_m(a)}\equiv 1 \pmod{m}, & \forall\, 0<i<\mathrm{ord}_m(a),\quad a^{\mathrm{ord}_m(a)}\not\equiv 1 \pmod{m}\\ b^{\mathrm{ord}_m(b)}\equiv 1 \pmod{m}, & \forall\, 0<j<\mathrm{ord}_m(b),\quad b^{\mathrm{ord}_m(b)}\not\equiv 1 \pmod{m}\end{cases}$.

$(ab)^{\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)}=a^{\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)}b^{\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)}\equiv 1\pmod{m}$.

记　$r=\mathrm{ord}_m(ab)$，则有　$r|\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)$，又有　$(\mathrm{ord}_m(a),\mathrm{ord}_m(b))=1$，则　r　的可取：$1,\mathrm{ord}_m(a),\mathrm{ord}_m(b),\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)$，下面展开讨论：

（1）当 $r=1$ 时：

$ab\equiv 1\pmod m$，（由3题知）则 $\mathrm{ord}_m(a)=\mathrm{ord}_m(b)$，又有$(\mathrm{ord}_m(a),\mathrm{ord}_m(b))=1$，则 $\mathrm{ord}_m(a)=\mathrm{ord}_m(b)=1$.

$r=\mathrm{ord}_m(ab)=\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)=1$成立.

（2）当 $r=\mathrm{ord}_m(a)$时：

$(ab)^r=(ab)^{\mathrm{ord}_m(a)}=b^{\mathrm{ord}_m(a)}\equiv 1\pmod m$，则 $\mathrm{ord}_m(b)\mid\mathrm{ord}_m(a)$，又有 $(\mathrm{ord}_m(a),\mathrm{ord}_m(b))=1$，则 $\mathrm{ord}_m(a)=\mathrm{ord}_m(b)=1$.

$r=\mathrm{ord}_m(ab)=\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)=1$成立.

（3）当 $r=\mathrm{ord}_m(b)$时，同（2）理，结论成立.

（4）当 $r=\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)$ 时，结论成立.

综上所示，$\mathrm{ord}_m(ab)=\mathrm{ord}_m(a)\cdot\mathrm{ord}_m(b)$，证毕.

5. 判断55,103的原根是否存在？若存在则求出其最小原根.

解 由原根存在的充要条件（2、4、奇素数的幂、2倍的奇素数的幂有原根）可知55的原根不存在，103的原根存在，下面求103的最小原根：

$\varphi(103)=102=2\times 3\times 17$，$\varphi(103)$有素因子 $q_1=2,\quad q_2=3,\quad q_3=17$，进而有：

$$\frac{\varphi(103)}{q_1}=51,\quad \frac{\varphi(103)}{q_2}=34,\quad \frac{\varphi(103)}{q_3}=6\,.$$

依次遍历计算：

$2^6\equiv 64\pmod{103},\quad 2^{34}\equiv 46\pmod{103},\quad 2^{51}\equiv 1\pmod{103}\longrightarrow$ 失败；

$3^6\equiv 8\pmod{103},\quad 3^{34}\equiv 1\pmod{103}\longrightarrow$ 失败；

$4^{51}=(2^{51})^2\equiv 1\pmod{103}\longrightarrow$ 失败；

$5^6\equiv 72\pmod{103},\quad 5^{34}\equiv 56\pmod{103},\quad 5^{51}\equiv 102\pmod{103}\longrightarrow$ 成功.

故5是103的最小原根.

6. 求出47的所有原根.

解 由原根存在的充要条件（2、4、奇素数的幂、2倍的奇素数的幂有原根）可知47的原根存在，下面求47的所有原根：

$\varphi(47)=46=2\times 23$，$\varphi(47)$有素因子 $q_1=2,\quad q_2=23$，进而有：$\dfrac{\varphi(47)}{q_1}=23,\quad \dfrac{\varphi(47)}{q_2}=2\,.$

依次遍历计算：

$2^2\equiv 4\pmod{47},\quad 2^{23}\equiv 1\pmod{47}\longrightarrow$ 失败；

$3^2\equiv 9\pmod{47},\quad 3^{23}\equiv 1\pmod{47}\longrightarrow$ 失败；

$4^{23}=(2^{23})^2\equiv 1\pmod{47}\longrightarrow$ 失败；

$5^2\equiv 25\pmod{47},\quad 5^{23}\equiv 46\pmod{47}\longrightarrow$ 成功.

当 t 遍历$\varphi(47)=46$的缩系：$1,3,5,7,9,11,13,15,17,19,21,25,27,29,31,33,35,37,39,41,43,45$ 时，5^t遍历47的原根，且共有 $\varphi(\varphi(47))=22$ 个原根.

$5^1 \equiv 5 \pmod{47}$　$5^3 \equiv 31 \pmod{47}$　$5^5 \equiv 23 \pmod{47}$　$5^7 \equiv 11 \pmod{47}$　$5^9 \equiv 40 \pmod{47}$

$5^{11} \equiv 13 \pmod{47}$　$5^{13} \equiv 43 \pmod{47}$　$5^{15} \equiv 41 \pmod{47}$　$5^{17} \equiv 38 \pmod{47}$　$5^{19} \equiv 10 \pmod{47}$

$5^{21} \equiv 15 \pmod{47}$　$5^{25} \equiv 22 \pmod{47}$　$5^{27} \equiv 33 \pmod{47}$　$5^{29} \equiv 26 \pmod{47}$　$5^{31} \equiv 39 \pmod{47}$

$5^{33} \equiv 35 \pmod{47}$　$5^{35} \equiv 29 \pmod{47}$　$5^{37} \equiv 20 \pmod{47}$　$5^{39} \equiv 30 \pmod{47}$　$5^{41} \equiv 45 \pmod{47}$

$5^{43} \equiv 44 \pmod{47}$　$5^{45} \equiv 19 \pmod{47}$

整理得47共有22个原根，分别为：$5,10,11,13,15,19,20,22,23,26,29,30,31,33,35,38,39,40,41,43,44,45.$

7. 已知2是19的原根，构造19的指数表并求解：

（1）$8x^4 \equiv 3 \pmod{19}$

（2）$5x^3 \equiv 2 \pmod{19}$

（3）$x^7 \equiv 1 \pmod{19}$

解　已知$g=2$ 是19的原根且 $\varphi(19)=18$，计算 $g^r \pmod{19}$　$(0 \le r \le 17)$，即：

$2^0 \equiv 1 \pmod{19}$　$2^1 \equiv 2 \pmod{19}$　$2^2 \equiv 4 \pmod{19}$　$2^3 \equiv 8 \pmod{19}$　$2^4 \equiv 16 \pmod{19}$

$2^5 \equiv 13 \pmod{19}$　$2^6 \equiv 7 \pmod{19}$　$2^7 \equiv 14 \pmod{19}$　$2^8 \equiv 9 \pmod{19}$　$2^9 \equiv 18 \pmod{19}$

$2^{10} \equiv 17 \pmod{19}$　$2^{11} \equiv 15 \pmod{19}$　$2^{12} \equiv 11 \pmod{19}$　$2^{13} \equiv 3 \pmod{19}$　$2^{14} \equiv 6 \pmod{19}$

$2^{15} \equiv 12 \pmod{19}$　$2^{16} \equiv 5 \pmod{19}$　$2^{17} \equiv 10 \pmod{19}$

构造19的指数表入下所示：

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	-	0	1	13	2	16	14	6	3	8
1	17	12	15	5	7	11	4	10	9	-

（1）由$8^{-1} \equiv 12 \pmod{19}$，原方程化简为：$x^4 \equiv 17 \pmod{19}$.

查表知：$\operatorname{ind}_g 17 = 10$ 且$d=(4,\varphi(19))=2 \mid \operatorname{ind}_g 17$，则该方程恰有2解.

原方程等价于：$4\operatorname{ind}_g x \equiv 10 \pmod{18}$，即$2\operatorname{ind}_g x \equiv 5 \pmod{9}$，得到 $\operatorname{ind}_g x \equiv 7 \pmod{9}$.

解得：$\operatorname{ind}_g x \equiv 7,16 \pmod{18}$.

查表得到：$x \equiv 5,14 \pmod{19}$.

（2）由$5^{-1} \equiv 4 \pmod{19}$，原方程化简为：$x^3 \equiv 8 \pmod{19}$.

查表知：$\operatorname{ind}_g 8 = 3$ 且$d=(3,\varphi(19))=3 \mid \operatorname{ind}_g 8$，则该方程恰有3解.

原方程等价于：$3\operatorname{ind}_g x \equiv 3 \pmod{18}$，即$\operatorname{ind}_g x \equiv 1 \pmod{6}$

解得：$\operatorname{ind}_g x \equiv 1,7,13 \pmod{18}$.

查表得到：$x \equiv 2,3,14 \pmod{19}$.

（3）查表知：$\mathrm{ind}_g 1 = 0$ 且$d = (7, \varphi(19)) = 1 \mid \mathrm{ind}_g 1$，则该方程恰有1解.

易知，解为：$x \equiv 1 \pmod{19}$.

8.（1）若g^k是m的原根，求证：g 是 m 的原根.

（2）若 p是一个以g为原根的奇素数，求证：$\mathrm{ind}_g(p-1) = \dfrac{p-1}{2}$.

证明　（1）由定理4.2.12：设 m 是大于2的整数，$\varphi(m)$的所有不同素因子是$q_1, q_2, \cdots, q_s$，则与m互素的正整数 g 是 m 的一个原根的充要条件是$g^{\frac{\varphi(m)}{q_i}} \not\equiv 1 \pmod{m}\quad i = 1, 2, \cdots, s$.

则对 $i = 1, 2, \cdots, s$有$(g^k)^{\frac{\varphi(m)}{q_i}} \not\equiv 1 \pmod{m}$，必有$g^{\frac{\varphi(m)}{q_i}} \not\equiv 1 \pmod{m}$（否则$(g^{\frac{\varphi(m)}{q_i}})^k \equiv 1 \pmod{m}$）

故g 是 m 的原根，证毕.

（2）由题易知，g 是 p 的原根，则有 $g^{\varphi(p)} = g^{p-1} \equiv 1 \pmod{p}$且 $\forall\, 0 < j < p-1,\ g^j \not\equiv 1 \pmod{p}$.

则 $g^{\frac{p-1}{2}} = -1 \pmod{p}$，即 $g^{\frac{p-1}{2}} = p-1 \pmod{p}$. 证毕.

9.设 p 是费马数 $F_n = 2^{2^n} + 1$的一个素因子，求证：

（1）$\mathrm{ord}_p(2) = 2^{n+1}$;

（2）p一定形如$2^{n+1}k + 1$.

（3）*(选做)当$n > 1$ 时，p一定形如$2^{n+2}t + 1$.

证明　（1）由题易知：$p \mid 2^{2^n} + 1$，即 $2^{2^n} \equiv -1 \pmod{p}$，则 $2^{2^{n+1}} = (2^{2^n})^2 \equiv 1 \pmod{p}$，那么$\mathrm{ord}_p(2) \mid 2^{n+1}$.

设$\mathrm{ord}_p(2) = 2^r (0 \le r \le n+1)$，对 r 的情况展开讨论：

i. 当 $r = 0$ 时：$\mathrm{ord}_p(2) = 1$，即 $2^1 \equiv 1 \pmod{p}$，显然不成立.

ii. 当 $0 < r < n+1$ 时：$\mathrm{ord}_p(2) = 2^r$，即 $2^{2^r} \equiv 1 \pmod{p} \quad \Leftrightarrow \quad (2^{2^{r-1}})^2 \equiv 1 \pmod{p}$，由 $\mathrm{ord}_p(2) = 2^r$ 可知：$2^{2^{r-1}} \not\equiv 1 \pmod{p}$，则 $2^{2^{r-1}} \equiv -1 \pmod{p}$，即 $p \mid 2^{2^{r-1}} + 1$，得到 $(2^{2^{r-1}} + 1, 2^{2^n} + 1) = p$，其中 $r - 1 < n$.

（由**P27定理1.5.5**：不同的两个费马数互素）推出矛盾，此时的 r 不满足.

综上，$r = n+1$，即$\mathrm{ord}_p(2) = 2^{n+1}$. 证毕.

（2）由欧拉定理可知：$2^{\varphi(p)} = 2^{p-1} \equiv 1 \pmod{p}$，则 $\mathrm{ord}_p(2) \mid p-1$，即$2^{n+1} \mid p-1$，则$p$一定形如$2^{n+1}k+1$. 证毕.

（3）计算Legendre符号：$\left(\dfrac{2}{p}\right)$.

由高斯引理书上已经推导过：$\left(\dfrac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = (-1)^{\frac{2^{n+1}k\cdot(2^{n+1}k+2)}{8}} = (-1)^{2^{n-1}k(2^n k+1)}$，由 $n > 1$ 知，$\left(\dfrac{2}{p}\right) = 1$.

使用欧拉判别条件计算：$\left(\dfrac{2}{p}\right) = 2^{\frac{p-1}{2}} = 2^{2^n k} = (2^{2^n})^k = (-1)^k$（其中最后一步推导用了 $2^{2^n} \equiv -1 \pmod{p}$）

则 $\exists\, t, k = 2t$. 证毕.

编程练习（基于C/C++）

1. 编程实现求解最小原根并基于最小原根构造指数表，效果如下图所示。

```
Please input n(n>0): 103
The min primitive root of 103: g=5
The ind_table of 103 based on g=5 is:
         0    1    2    3    4    5    6    7    8    9
    0    -    0   44   39   88    1   83    4   30   78
    1   45   61   25   72   48   40   74   70   20   80
    2   89   43    3   24   69    2   14   15   92   86
    3   84   57   16  100   12    5   64   93   22    9
    4   31   50   87   77   47   79   68   85   11    8
    5   46    7   58   97   59   62   34   17   28   98
    6   26   36  101   82   60   73   42   13   56   63
    7   49   67    6   33   35   41   66   65   53   18
    8   75   54   94   38   29   71   19   23   91   99
    9   21   76   10   96   27   81   55   32   52   37
   10   90   95   51    -    -    -    -    -    -    -
```

```
Please input n(n>0): 169
The min primitive root of 169: g=2
The ind_table of 169 based on g=2 is:
         0    1    2    3    4    5    6    7    8    9
    0    -    0    1  124    2    9  125  107    3   92
    1   10  103  126    -  108  133    4  146   93   65
    2   11   75  104  130  127   18    -   60  109   40
    3  134   21    5   71  147  116   94  151   66    -
    4   12   85   76  122  105  101  131   63  128   58
    5   19  114    -  120   61  112  110   33   41   35
    6  135  140   22   43    6    -   72   37  148   98
    7  117  137   95   51  152  142   67   54    -   24
    8   13   28   86   45   77  155  123    8  106   91
    9  102    -  132  145   64   74  129   17   59   39
   10   20   70  115  150    -   84  121  100   62   57
   11  113  119  111   32   34  139   42    -   36   97
   12  136   50  141   53   23   27   44  154    7   90
   13    -  144   73   16   38   69  149   83   99   56
   14  118   31  138    -   96   49   52   26  153   89
   15  143   15   68   82   55   30    -   48   25   88
   16   14   81   29   47   87   80   46   79   78    -
```

```cpp
#include<iostream>
#include<unordered_map>
#include<vector>
#include<iomanip>
using namespace std;
//Unique Factorization Theorem
unordered_map<int, int>* numDecompose(int n)
{
    unordered_map<int, int>* nums = new unordered_map<int, int>();
    while (true)
    {
        int i = 2;
        while (i <= n)
        {
            if (n % i == 0)
```

```
            {
                auto iter = nums->find(i);
                if (iter == nums->end())
                    nums->emplace(i, 1);
                else
                    iter->second++;
                n /= i;
                break;
            }
            i++;
        }
        if (n == 1)break;
    }
    return nums;
}
//pow_mod: x^y mod m
int pow_mod(int x, int y, int m)
{
    int rst = 1;
    while (y > 0)
    {
        if (y & 1)
        {
            rst *= x;
            rst %= m;
        }
        x *= x;
        x %= m;
        y >>= 1;
    }
    return rst;
}
//iff: 2,4,p^l,2p^l
bool hasPrimitiveRoot(const unordered_map<int, int>* nums)
{
    int count = nums->size();
    auto pos2 = nums->find(2);
    //2,4,p^l
    bool flag1 = count == 1 && (pos2 == nums->end() || pos2->second == 1 || pos2->second == 2);
    //2p^l
    bool flag2 = count == 2 && pos2 != nums->end() && pos2->second == 1;
    return flag1 || flag2;
}
//phi(n)
int phi(int n, const unordered_map<int, int>* nums)
{
    for (auto iter = nums->begin(); iter != nums->end(); iter++)
    {
        n /= iter->first;
        n *= iter->first - 1;
    }
    return n;
}
//calculate min primitive root
int calcMinPrimitiveRoot(int n, int phi_n)
{
    int min_g = 0;
    unordered_map<int, int>* phi_n_nums = numDecompose(phi_n);
    vector<int>powerNum;
    for (auto iter = phi_n_nums->begin(); iter != phi_n_nums->end(); iter++)
```

```cpp
        powerNum.push_back(phi_n / iter->first);
    for (int i = 2; i < n; i++)
    {
        bool flag = true;
        for (int j = 0; j < powerNum.size(); j++)
        {
            if (pow_mod(i, powerNum[j], n) == 1)
            {
                flag = false;
                break;
            }
        }
        if (flag)
        {
            min_g = i;
            break;
        }
    }
    delete phi_n_nums;
    return min_g;
}
//make ind_table and print
void makeIndTableAndPrint(int n, int phi_n, int g)
{
    unordered_map<int, int>ind_map;
    for (int r = 0; r < phi_n; r++)
    {
        ind_map.emplace(pow_mod(g, r, n), r);
    }
    //print
    cout << "The ind_table of " << n << " based on g=" << g << " is: " << endl;
    int tens = n / 10;
    for (int i = 0; i <= tens + 1; i++)
    {
        for (int j = 0; j <= 10; j++)
        {
            if (i == 0 && j == 0)
            {
                cout << setw(5) << " ";
            }
            else if (i * j == 0)
            {
                cout << setw(5) << i + j - 1;
            }
            else
            {
                int t = 10 * (i - 1) + j - 1;
                if (ind_map.find(t) != ind_map.end())
                    cout << setw(5) << ind_map.find(t)->second;
                else
                    cout << setw(5) << "-";
            }
        }
        cout << endl;
    }
}
int main()
{
    int n;
    cout << "Please input n(n>0): ";
    cin >> n;
```

```
    unordered_map<int, int>* nums = numDecompose(n);
    if (hasPrimitiveRoot(nums))
    {
        int phi_n = phi(n, nums);
        int min_g = calcMinPrimitiveRoot(n, phi_n);
        cout << "The min primitive root of " << n << ": " << "g=" << min_g << endl;
        makeIndTableAndPrint(n, phi_n, min_g);
    }
    else
    {
        cout << "N has no primitive root! No ind_table! " << endl;
    }
    delete nums;
    return 0;
}
```