算法第一周作业

1.证明gcd(m,n) = gcd(n, m mod n)

> 证明：
>
> 假设m和n的最大公约数是d，
> 则存在唯一的正整数k1，使m = k1 * d，
> 同理，存在唯一的正整数k2，使n = k2 * d，
> 且k1和k2互质，
> 则存在唯一的正整数k，使m = k * n + r，
> 即r = m mod n，
> 则r = m - k * n = (k1-k2*k*) * d
> 而(k1-k2*k)也是一个正整数，
> 所以(m mod n)也是d的倍数，
> 即证明了m和n的公约数就是n和m mod n的公约数

2.设计计算$\sqrt{n}$的算法，n为任意整数，除了赋值与比较运算，只能用到基本四则运算

> (1)判断n是否是一个正整数，若是，进入(2)；否则，返回非法输入
>
> (2)取下界为min=1，上界为max=n，令ans=(max+min)/2
>
> (3)判断(max-min<=1)是否为真，若是，进入(4)；否则，进入(5)
>
> (4)判断(max*max==n)是否为真，若是，返回max；否则，返回min
>
> (5)判断(ans*ans>n是否为真)，若是，max=ans，ans=(max+min)/2，进入(3)；否则，min=ans，ans=(max+min)/2，进入(3)
>
> 时间复杂性为O(logn)
>
> C++代码如下：

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cout<<"请输入一个正整数：";
    cin>>n;
    if(cin.fail()){
        cout<<"输入错误"<<endl;
    }
    else if(n<0){
        cout<<"输入的不是正数"<<endl;
    }
    else{
        int mx=n,mn=,ans=(mx+mn)/2;
        while(mx-mn>1){
            if(ans*ans>n)
                mx=ans;
            else
                mn=ans;
            ans=(mx+mn)/2;
        }
        if(mx*mx==n)
            cout<<"向下取整的平方根为："<<mx<<endl;
        else
            cout<<"向下取整的平方根为："<<mn<<endl;
    }
    return 0;
}
```

3.证明主定理

> 证明:
>
> $T(n) = aT(\frac{n}{b}) + O(n^d)$
>
> $= a[aT(\frac{n}{b^2}) + O((\frac{n}{b})^d)] + O(n^d)$
>
> $= a^kT(\frac{n}{b^k}) + \sum_{i=0}^{k-1} a^iO((\frac{n}{b^i})^d)$
>
> $= a^kO(1) + O(n^d(\sum_{i=0}^{k-1} (\frac{a}{b^d})^i))$
>
> 其中，$k = log_bn$

> 令 $q = \frac{a}{b^n}$
>
> (1)当q=1，即$a = b^n$ 时
>
> $T(n) = O(n^d * k)$
>
> $= O(n^dlog_bn)$
>
> $= O(n^dlogn)$
>
> (2)当q<1，即$a < b^n$ 时
>
> $T(n) = O(n^d * \frac{1}{1-q})$

> $= O(n^d)$
>
> (3)当q>1，即$a > b^n$ 时
>
> $T(n) = O(n^d * \frac{1-q^{log_b n}}{1-q})$
>
> $= O(n^d * (\frac{a}{b^d})^{log_b n})$
>
> $= O(n^{log_b a})$

4.1-6，1-7

1-6

> 1.$\log n^2 = \Theta(\log n + 5)$
>
> 因为$f(n) = 2\log n$，与g(n)阶数相同
>
> 2.$\log n^2 = O(\sqrt{n})$
>
> 因为指数是正数的幂函数的阶数大于对数函数的阶数
>
> 3.$n = \Omega(\log^2 n)$
>
> 因为指数是正数的幂函数的阶数大于对数函数平方的阶数
>
> 4.$n\log n + n = \Omega(\log n)$
>
> 因为$f(n) = n\log n + n = \Theta(n\log n)$，而$g(n) = \Theta(logn)$
>
> 5.$10 = \Theta(\log 10)$
>
> 因为两个常数函数的阶数相同
>
> 6.$\log^2 n = \Omega(\log n)$
>
> 因为对数函数平方的阶数大于对数函数的阶数
>
> 7.$2^n = \Omega(100n^2)$
>
> 因为指数函数的阶数大于指数是正数幂函数的阶数
>
> 8.$2^n = O(3^n)$
>
> 因为对于底数大于1的指数函数，底数越大，阶数越大

1-7

> 证明：
>
> $$\lim_{n\to\infty} \frac{n!}{n^n} = \lim_{n\to\infty} (\frac{1}{n} * \frac{2}{n} * ... > * \frac{n-1}{n} * \frac{n}{n}) = 0$$
>
> 所以，对于任意的$c > 0$，存在$n_0 > 0$，使得对于任意的$n \geq n_0$，$0 \leq f(n) < c * g(n)$