

牛客多校第七场

B

GCD的递归。比较难的C语言练习。

```
#include<stdio.h>

long long int t, tmp, ans[20005], top = 1;

long long gcd(long long a, long long b)
{
    return b ? gcd(b, a % b) : a;
}

void solve(long long int n, long long int m, long long int res)
{
    if(n > m)
    {
        tmp = m;
        m = n;
        n = tmp;
    }
    if(res == 0)
    {
        return;
    }
    long long int k = m - m % n;
    while(k--)
    {
        ans[top++] = n;
    }
    solve(n, m % n, n * (m % n));
    return;
}

int main()
{
    scanf("%d", &t);
    while(t--)
    {
        long long n, m;
        top = 1;
        scanf("%lld%lld", &n, &m);
        solve(n, m, n * m);
        long long l = n + m - gcd(n, m);
        printf("%lld\n", l);
    }
}
```

```
int i;
for(i=1;i<=l;i++)
{
    printf("%lld ",ans[i]);
}
printf("\n");
}
```

D

水。

```
#include<stdio.h>
#include<math.h>

int main()
{
    int k;
    scanf("%d",&k);
    while(k--)
    {
        int n;
        scanf("%d",&n);
        if(n==1 || n==24)
        {
            printf("Fake news!\n");
        }
        else
        {
            printf("Nobody knows it better than me!\n");
        }
    }
    return 0;
}
```

H

数论分块。注意先加上模再取模。

```
#include<stdio.h>

long long int n, k, r, l, cnt, cnt1;

int main()
```

```

{
    scanf("%lld%lld", &n, &k);
    if(k > n)
    {
        cnt1 = (cnt1 + k - n + 1000000007) % 1000000007;
    }
    k=n<k?n:k;
    for(l = 1ll; l <= k; l = r + 1ll)
    {
        if(n / l)
        {
            r = n/(n/l);
        }
        else
        {
            r = k;
        }
        r=r<k?r:k;
        cnt = (cnt + (r - l + 1ll) * (n / l) + 1000000007) % 1000000007;
        if(n % r == 0ll)
        {
            cnt1 = (cnt1 + r - l + 1000000007) % 1000000007;
        }
        else
        {
            cnt1 = (cnt1 + r - l + 1ll + 1000000007) % 1000000007;
        }
    }
    cnt = (2ll * cnt - n + 1000000007) % 1000000007;
    printf("%lld\n", (cnt + cnt1 + 1000000007) % 1000000007);
}

```

A

以下是补题。

这个题数据量较小，甚至可以打表。

```

#include<stdio.h>
#include<string.h>
#include<math.h>

int f[9][2*250][2*250], n, r;
int Ans[20][50];

int main()
{
    int T;
    scanf("%d",&T);

```

```
memset(Ans,255,sizeof(Ans));
while (T--)
{
    scanf("%d%d", &n, &r);
    if (Ans[n][r]!=-1)
    {
        printf("%d\n",Ans[n][r]);
        continue;
    }
    int i;
    for(i=0; i<=n; ++i)
    {
        int j;
        for(j=250-n*r; j<=250+n*r; ++j)
        {
            int k;
            for(k=250-n*r; k<=250+n*r; ++k)
            {
                f[i][j][k]=-1;
            }
        }
    }
    f[0][250][250]=0;
    for(i=-r; i<=r; ++i)
    {
        int j;
        for(j=-r; j<=r; ++j)
        {
            if(i*i+j*j<=r*r)
            {
                f[1][250+i][250+j] = 0;
            }
        }
    }
    for(i=2; i<=n; ++i)
    {
        int j;
        for(j=-r; j<=r; ++j)
        {
            int k = trunc(sqrt(r*r-j*j));
            int s1;
            for(s1=250-(i-1)*r; s1<=250+(i-1)*r; ++s1)
            {
                int s2;
                for(s2=250-(i-1)*r; s2<=250+(i-1)*r; ++s2)
                {
                    int z = f[i-1][s1][s2]; int t1 = s1 - 250; int t2 =
s2 - 250;

                    if (z==-1) continue;
                    f[i][s1+j][s2+k]
```

```

=f[i][s1+j][s2+k]>(z+(t1*t1+t2*t2+z)/(i-1)-2*(j*t1+k*t2)+(i-1)*(j*j+k*k))?f
[i][s1+j][s2+k):(z+(t1*t1+t2*t2+z)/(i-1)-2*(j*t1+k*t2)+(i-1)*(j*j+k*k));
        f[i][s1+j][s2-k] =f[i][s1+j][s2-
k]>(z+(t1*t1+t2*t2+z)/(i-1)-2*(j*t1-k*t2)+(i-1)*(j*j+k*k))?f[i][s1+j][s2-
k):(z+(t1*t1+t2*t2+z)/(i-1)-2*(j*t1-k*t2)+(i-1)*(j*j+k*k));
    }
    }
    }
    }
    int ans = 0;
    for(i=250-n*r; i<=250+n*r; ++i)
    {
        int j;
        for(j=250-n*r; j<=250+n*r; ++j)
        {
            ans =ans>f[n][i][j]?ans:f[n][i][j];
        }
    }
    printf("%d\n", ans);
    Ans[n][r]=ans;
}
return 0;
}

```

```

#include<stdio.h>

int T;
long long mod;
long long norings[5007], GraphCnt[5007], EdgeCnt[5007], f[5007], g[5007];
long long SS[5007][5007];

int power(int x, int times)
{
    return SS[x][times];
}

long long C[5007][5007];

int main()
{
    scanf("%d%d", &T, &mod);
    int i;
    for(i = 1; i < 5007; i++)
    {
        SS[i][0] = 1;
        int j;
        for(j = 1; j < 5007; j++)

```

```
        SS[i][j] = (long long)SS[i][j - 1] * i % mod;
    }
    for(i = 0; i < 5007; i++)
    {
        C[i][0] = C[i][i] = 1;
        int j;
        for(j = 1; j < i; j++)
        {
            C[i][j] = C[i - 1][j - 1] + C[i - 1][j] ;
            if (C[i][j] >= mod) C[i][j]-=mod;
        }
    }
    norings[1] = norings[2] = 1;
    for(i = 3; i < 5007; i++)
    {
        norings[i] = power(i, i - 2);
    }
    for(i = 1; i < 5007; i++)
    {
        GraphCnt[i] = norings[i];
    }
    for(i = 2; i < 5007; i++)
    {
        EdgeCnt[i] = 0;
        int j;
        for(j = 1; j <= i - 1; j++)
        {
            EdgeCnt[i] = (EdgeCnt[i] + C[i - 2][j - 1] * power(i - 1, i - j
- 1) % mod * i % mod * (j * j) % mod) % mod;
        }
    }
    f[0] = 1;
    for(i = 1; i < 5007; i++)
    {
        int j;
        for(j = 1; j <= i; j++)
        {
            f[i] = (f[i] + GraphCnt[j] * f[i - j] % mod * C[i - 1][j - 1])
% mod;
            g[i] = (g[i] + C[i - 1][j - 1] * (EdgeCnt[j] * f[i - j] % mod +
GraphCnt[j] * g[i - j] % mod)) % mod;
        }
    }
    while (T--)
    {
        int n;
        scanf("%d",&n);
        printf("%lld\n",g[n]);
    }
    return 0;
```

```
}

```

J

见[小型代码分析系统的实现方式](#)

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E7%89%9B%E5%AE%A2%E5%A4%9A%E6%A0%A1%E7%AC%ACE4%B8%83%E5%9C%BA>

Last update: 2020/08/08 11:26