

类欧几里得算法

算法思想

我们设 $f(a,b,c,n)=\sum_{i=0}^n \lfloor \frac{ai+b}{c} \rfloor$

其中 a,b,c,n 为常数，我们需要一个 $O(\log n)$ 的算法。

如果 $a \geq c$ 或者 $b \geq c$ 我们可以将 a,b 对 c 取模来化简问题：

$$f(a,b,c,n)=\sum_{i=0}^n \lfloor \frac{ai+b}{c} \rfloor$$

$$=\sum_{i=0}^n \lfloor \frac{\lfloor \frac{a}{c} \rfloor c + a \bmod c + \lfloor \frac{b}{c} \rfloor c + b \bmod c}{c} \rfloor$$

$$=\lfloor \frac{n(n+1)}{2} \rfloor \lfloor \frac{a}{c} \rfloor + (n+1) \lfloor \frac{b}{c} \rfloor + \sum_{i=0}^n \lfloor \frac{(a \bmod c) + (b \bmod c)}{c} \rfloor$$

$$=\lfloor \frac{n(n+1)}{2} \rfloor \lfloor \frac{a}{c} \rfloor + (n+1) \lfloor \frac{b}{c} \rfloor + f(a \bmod c, b \bmod c, c, n)$$

这样我们就将前两个参数控制到一定比第三个参数小的形式了。

$$\sum_{i=0}^n \lfloor \frac{ai+b}{c} \rfloor = \sum_{i=0}^n \sum_{j=0}^{\lfloor \frac{ai+b}{c} \rfloor - 1} 1$$

然后我们交换和号：

$$\sum_{j=0}^{\lfloor \frac{an+b}{c} \rfloor - 1} \sum_{i=0}^n [j < \lfloor \frac{ai+b}{c} \rfloor]$$

对于里面的式子，我们可以变换一下：

$$j < \lfloor \frac{ai+b}{c} \rfloor \iff j+1 \leq \lfloor \frac{ai+b}{c} \rfloor \iff j+c \leq ai+b \iff j+c-b-1 \leq ai \iff \lfloor \frac{j+c-b-1}{a} \rfloor \leq i$$

$$\text{这样我们设 } m = \lfloor \frac{an+b}{c} \rfloor$$

原式变为 $f(a,b,c,n) = \sum_{j=0}^{m-1} \sum_{i=0}^n [i \leq \lfloor \frac{j+c-b-1}{a} \rfloor] = \sum_{j=0}^{m-1} (n - \lfloor \frac{j+c-b-1}{a} \rfloor) = nm - f(c, c-b-1, a, m-1)$ 然后第一个参数又比第三个大了，就一直取模这样，类似于求最大公约数。

算法实现

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
```

```
const ll mod=998244353,base=233;

ll Hash(int a,int b,int c,int n) {
    return (((((ll)a*base%mod)+b)*base%mod+c)*base%mod+n)%mod;
}
unordered_map<ll,int> F;

ll f(int a,int b,int c,int n) {
    if(!a) return (ll)b/c*(n+1)%mod;
    ll tmp=Hash(a,b,c,n);
    if(F.find(tmp)!=F.end()) return F[tmp];
    if(a>=c||b>=c) return
F[tmp]=(((ll)n*(n+1)/2%mod*(a/c)%mod+((ll)n+1)*(b/c)%mod)%mod+f(a%c,b%c,c,n))%mod;
    int m=((ll)a*n+b)/c;
    return F[tmp]=((ll)n*m%mod-f(c,c-b-1,a,m-1)+mod)%mod;
}
int n,a,b,c;
int main() {
    int t;
    scanf("%d",&t);
    while(t--) {
        scanf("%d %d %d %d",&n,&a,&b,&c);
        printf("%lld\n",f(a,b,c,n));
    }
    return 0;
}
```

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:%E7%8E%8B%E6%99%BA%E5%BD%AA:%E7%B1%BB%E6%AC%A7%E5%87%A0%E9%87%8C%E5%BE%97%E7%AE%97%E6%B3%95&rev=1629080800

Last update: 2021/08/16 10:26

