整除分块

人员

叶乐山、于飞岚、陈瀚霄、方冠霖、张皓宁、卢炫佑 到课, 曹承贤 线上

本周作业

https://vjudge.net/contest/764809

课堂表现

同学们这节课听讲都很认真, 课上做题表现也非常好

课堂内容

P9611 [CERC2019] Zeldain Garden

整除分块模板题, 求 n/1 + n/2 + n/3 + ... + n/n

分成 sqrt(n) 块, 每块 O(1) 求即可

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

LL calc(LL x) {
  LL res = 0;
  for (LL i = 1; i <= x; ) {
    LL v = x / i;
    LL j = x / v;
    res += (LL)v * (j-i+1);
    i = j + 1;
  }
  return res;
}

int main()
{
  LL l, r; cin >> l >> r;
  cout << calc(r) - calc(l-1) << endl;
  return 0;
}
```

P2424 约数和

求 (n/1) * 1 + (n/2) * 2 + (n/3) * 3 + ... + (n/n) * n

分成 sqrt(n) 块, 每块需要用一下等差公式求和, 然后 O(1) 求

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

LL get_sum(int i, int j) { return ((LL)i+j)*(j-i+1)/2; }

LL calc(int x) {
  LL res = 0;
  for (int i = 1; i <= x; ) {
    int v = x / i;
    int j = x / v;
    res += get_sum(i,j)*v;
    i = j+1;
  }
  return res;
}

int main()
{
  int l, r; cin >> l >> r;
  cout << calc(r) - calc(l-1) << endl;
  return 0;
}
```

P2261 [CQOI2007] 余数求和

k % i -> k - (k/i)*i

然后就转变成了上一道题

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

LL get_sum(int l, int r) { return ((LL)l+r)*(r-l+1)/2; }

int main()
{
  int n, k; cin >> n >> k;
  LL res = (LL)n * k;
  for (int i = 1; i <= n; ) {
    int v = k / i;
```

```
      if (!v) break;
      int j = min(k/v, n);
      res -= (LL)v * get_sum(i,j);
      i = j + 1;
    }
    cout << res << endl;
    return 0;
  }
```

CF1561D1 Up the Strip (simplified version)

dp + 整除分块

f[i] 需要转移到 f[i-1], f[i-2], ..., f[1], f[i/2], f[i/3], ..., f[i/i]

针对减法的转移, 可以利用一个 sum 进行 O(1) 的转移

针对除法的转移, 可以利用整除分块, 进行 O(sqrt(n)) 的转移

时间复杂度: O(n*sqrt(n))

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int f[maxn];

int main()
{
  int n, mod; cin >> n >> mod;

  int sum = 0;
  f[n] = 1;
  for (int i = n; i >= 1; --i) {
    f[i] = (f[i] + sum) % mod;
    for (int l = 2; l <= i; ) {
      int v = i / l;
      int r = i / v;
      f[v] = (f[v] + 1LL*f[i]*(r-l+1)%mod) % mod;
      l = r+1;
    }
    sum = (sum + f[i]) % mod;
  }
  cout << f[1] << endl;
  return 0;
}
```

CF1561D2 Up the Strip

dp + 调和级数

针对减法的转移, 跟上个题一致

针对除法的转移, 考虑 f[i]

可以从 f[i*2] ~ f[(i+1)*2-1] 转移而来

可以从 f[i*3] ~ f[(i+1)*3-1] 转移而来

可以从 f[i*4] ~ f[(i+1)*4-1] 转移而来

...

每一段转移可以用前缀和优化, 实现 O(1) 转移

针对 i 来说, 最多有 n/i 段

时间复杂度: n * logn

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 4e6 + 5;
int mod;
int f[maxn], p[maxn];

int get_sum(int l, int r) { return (p[l]-p[r+1]+mod) % mod; }

int main()
{
  int n; cin >> n >> mod;

  f[n] = 1;
  for (int i = n; i >= 1; --i) {
    f[i] = (f[i] + get_sum(i+1,n)) % mod;

    for (int k = 2; k <= n/i; ++k) {
      int l = i*k, r = min((i+1)*k-1, n);
      f[i] = (f[i] + get_sum(l,r)) % mod;
    }
    p[i] = (p[i+1] + f[i]) % mod;
  }
  cout << f[1] << endl;
  return 0;
}
```