知识精炼（一）

主讲人：邓哲也

Codeforces 958C3. Encryption

给定一个长度为 N 的正整数序列，请你把它分成 K 段，使得每一段的和对 P 取模之后相加的总和最小。

$N \leqslant 500000$，$2 \leqslant K, P \leqslant 100$

样例： （输出：6）

4 3 10

3 4 7 2

Codeforces 958C3.Encryption

先考虑简单的dp：

用 sum[i] 表示前 i 个数的和。

用 f[i][j] 表示前 i 个数分成 j 段得到的最小值。

转移方程：

$$f[i][j] = \min\{f[k][j-1]+(sum[i]-sum[k]) \% p \mid 0 \leq k < i\}$$

这样的时间复杂度是 O(N^2 K)

Codeforces 958C3. Encryption

用 sum[i] 表示前 i 个数的和对 P 取模的结果。

这样区间 [x, y] 的和对 P 取模的答案就是：

```
if sum[x] ≤ sum[y]:    sum[y] - sum[x]
else:                  sum[y] - sum[x] + P
```

Codeforces 958C3. Encryption

回到转移方程：

$f[i][j] = \min\{f[k][j-1]+(sum[i]-sum[k]+P)\%P \mid 0 \leqslant k < i\}$

如果 $sum[k] \leqslant sum[i]$：

$f[i][j] = \min\{f[k][j-1]+sum[i]-sum[k] \mid 0 \leqslant k < i\}$

$= \min\{f[k][j-1]-sum[k] \mid 0 \leqslant k < i\} + sum[i]$

否则：

$f[i][j] = \min\{f[k][j-1]+sum[i]-sum[k]+P \mid 0 \leqslant k < i\}$

$= \min\{f[k][j-1]-sum[k]+P \mid 0 \leqslant k < i\} + sum[i]$

Codeforces 958C3.Encryption

先来看 $\min\{f[k][j-1]-sum[k] \mid 0 \leqslant k < i\}$

要满足 sum[k]≤sum[i] 且 0≤k<i，我们只需要从小到大枚举 i，这样自然就满足了后者。

也就是每次把 k < i 的 (f[k][j-1]-sum[k]) 插入到下标为 sum[k] 的位置上。

查询的时候在下标区间[0, sum[i]]中查询最小值即可。

Codeforces 958C3. Encryption

min{f[k][j-1]-sum[k]+P | 0≤k<i}

同理用另一个数据结构把（f[k][j-1]-sum[k]+P）插入到下标为 sum[k] 的位置上。

查询的时候在下标区间[sum[i]+1, P)中查询最小值。

单点修改 + 前缀区间查询 ➔ 树状数组！

时间复杂度 O(NK log P)

Codeforces 958C3. Encryption

```
int bit[2][105][105];
void add(int i, int j, int x, int v){
      // 修改的范围是[0, P]，需要映射到[1, P + 1]
      for(x ++;x <= P + 1; x += x & -x)
            bit[i][j][x] = min(bit[i][j][x], v);
}
```

Codeforces 958C3. Encryption

```
int bit[2][105][105];
int ask(int i, int j, int x){
      int ans = 1e9;
      for(x ++;x;x -= x & -x)
            ans = min(ans, bit[i][j][x]);
      return ans;
}
```

Codeforces 958C3. Encryption

```
int main(){
    scanf("%d%d%d", &n, &k, &p);
    for(int i = 1;i <= n;i ++){
        scanf("%d", &sum[i]);
        sum[i] = (sum[i - 1] + sum[i]) % p;
    }
    memset(bit, 0x3f, sizeof(bit));
    memset(f, 0x3f, sizeof(f));
    add(0, 0, 0, 0);
    add(1, 0, 0, 0);
```

Codeforces 958C3. Encryption

```
    for(int i = 1;i <= n;i ++){
        for(int j = 1;j <= min(i, k);j ++)
            f[j] = min(ask(0,j-1,sum[i]),ask(1,j-1,
                        p-sum[i]))+sum[i];
        for(int j = 1;j <= min(i, k);j ++){
            if (f[j] >= 1e9) continue;
            add(0, j, sum[j], f[j] - sum[j]);
            add(1, j, p - sum[j], f[j] - sum[j] + p);
        }
    }
    printf( “%d\n” , f[k]);
    return 0;
}
```

下节课再见