杂题混练 + 质因数分解

人员

蔡云翔、石宇嫿、李佳声、胡赫轩、崔嘉睿、穆鹏宇、程晟泰、梁钰涵、刘浩、纪昀琨、张彧韶、于潇涵、周子航 到课

作业检查

上周作业链接：https://vjudge.net/contest/655631

密码：code@123

梁钰涵、穆鹏宇 做了 4 道题

胡赫轩 做了 3 道题

纪昀琨、程晟泰 做了 1 道题

其他同学 未做

作业

https://vjudge.net/contest/655631, 上周 5 道作业题要求大家补完

https://www.luogu.com.cn/contest/200616, 课上 K 题要求大家补完

https://vjudge.net/contest/658739, 课后作业 5 道题要求大家课后进行思考尝试

课堂表现

整体课堂纪律都很好，同学们目前就是补题不积极，课下要多花时间补题

课堂内容

CF1485C Floor and Mod

a = i * b + i

枚举 i, 之后 b_min = i+1, b_max = min(y, (x-i)/i)

i*i <= x -> i <= sqrt(x)

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

using namespace std;

typedef long long LL;

void solve() {
```

```
  int x, y; cin >> x >> y;
  LL res = 0;
  for (int i = 1; i <= y; ++i) {
    int b_min = i+1, b_max = min((x-i) / i, y);
    if (b_min > b_max) break;
    res += b_max - b_min + 1;
  }

  cout << res << endl;
}

int main()
{
  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

CF417D Cunning Gena 1900

先按照每个元素的 k 进行排序，之后进行状压即可

一开始先根据 ki 从小到大进行排序

f[i][j]: 只考虑前 i 个人时，要达到状态 j，最少花多少钱

答案: f[i][(1<<m)-1] + ki*b

转移: 第 i 个人花 xi 的钱，他能解决的问题是 valuei

f[i][j|valuei] = min(f[i-1][j|valuei], f[i-1][j] + xi)

初值: memset(f, 0x3f, sizeof(f)), f[0][0] = 0

炸空间，用滚动数组优化

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

using namespace std;

typedef long long LL;
const int N = 100 + 5, M = 21;
const LL inf = 0x3f3f3f3f3f3f3f3f;
struct node {
  int x, k, val;
  bool operator < (const node& p) const { return k < p.k; }
} w[N];
LL f[1<<M];
```

```
int main()
{
  int n, m, b; cin >> n >> m >> b;
  for (int i = 1; i <= n; ++i) {
    int x, k, m; cin >> x >> k >> m;
    int val = 0;
    while (m -- ) { int t; cin >> t; --t; val |= (1<<t); }
    w[i] = {x, k, val};
  }
  sort(w+1, w+n+1);

  memset(f, 0x3f, sizeof(f));
  f[0] = 0;
  LL res = inf;
  for (int i = 1; i <= n; ++i) {
    for (int j = 0; j < (1<<m); ++j) {
      f[j|w[i].val] = min(f[j|w[i].val], f[j] + w[i].x);
    }
    res = min(res, f[(1<<m)-1] + (LL)w[i].k*b);
  }

  cout << (res==inf ? -1 : res) << endl;
  return 0;
}
```

CF1379C Choosing flowers

最终只会选择一个 b 一直选，可以枚举每次最后选择哪个 b

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
struct node {
  int a, b;
  bool operator < (const node& p) const { return a < p.a; }
} w[maxn];
LL a[maxn], p[maxn];

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

void solve() {
  int m, n; cin >> m >> n;
  for (int i = 0; i <= n+2; ++i) p[i] = 0;
  for (int i = 1; i <= n; ++i) cin >> w[i].a >> w[i].b;
  sort(w+1, w+n+1);
  for (int i = 1; i <= n; ++i) {
    a[i] = w[i].a; p[i] = p[i-1] + a[i];
  }
```

```
  LL res = 0;
  for (int i = 1; i <= n; ++i) {
    int p = upper_bound(a+1, a+n+1, w[i].b) - a;
    if (n-p+1 >= m) {
      res = max(res, get_sum(n-m+1, n));
    } else {
      if (i >= p) res = max(res, get_sum(p, n)+(LL)w[i].b*(m-(n-p+1)));
      else res = max(res, get_sum(p,n) + w[i].a + (LL)w[i].b*(m-(n-p+1)-1));
    }
  }

  cout << res << endl;
}

int main()
{
  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

CF7D Palindrome Degree

hash + dp

维护一个前缀 hash, 再维护一个后缀 hash 即可

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

using namespace std;

typedef long long LL;
typedef unsigned long long ULL;
const int maxn = 5e6 + 5;
const int P = 131;
char s[maxn];
ULL p[maxn], pre_h[maxn], suf_h[maxn];
int f[maxn];

void init(int n) {
  p[0] = 1;
  for (int i = 1; i <= n; ++i) p[i] = p[i-1]*P;
  pre_h[0] = 1;
  for (int i = 1; i <= n; ++i) pre_h[i] = pre_h[i-1]*P + s[i];
  suf_h[n+1] = 1;
  for (int i = n; i >= 1; --i) suf_h[i] = suf_h[i+1]*P + s[i];
}

ULL get_hash(int l, int r, bool is_pre) {
  if (is_pre) return pre_h[r] - pre_h[l-1]*p[r-l+1];
```

```
  return suf_h[l] - suf_h[r+1]*p[r-l+1];
}

int main()
{
  ios::sync_with_stdio(false);
  cin.tie(0);
  cin >> (s+1);
  int n = strlen(s+1);
  init(n);

  LL res = 0;
  for (int i = 1; i <= n; ++i) {
    if (i == 1) f[i] = 1;
    else {
      int len = i / 2;
      if (get_hash(1, len, true) == get_hash(i-len+1, i, false)) f[i] = f[len]+1;
      else f[i] = 0;
    }
    res += f[i];
  }
  cout << res << endl;
  return 0;
}
```

CF1276C Beautiful Rectangle

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

using namespace std;

struct node {
  int val, c;
  bool operator < (const node& p) const { return c < p.c; }
};

const int maxn = 4e5 + 5;
vector<int> w[maxn];

int main()
{
  ios::sync_with_stdio(false);
  cin.tie(0); cout.tie(0);

  int n; cin >> n;
  map<int, int> mp;
  for (int i = 1; i <= n; ++i) {
    int x; cin >> x; ++mp[x];
  }
  vector<node> vec;
  for (auto it : mp) vec.push_back({it.first, it.second});
```

```
  sort(vec.begin(), vec.end()); reverse(vec.begin(), vec.end());

  int res = 0, x = 0, y = 0;
  for (int i = 1; i*i <= n; ++i) {
    int cnt = 0;
    for (node it : vec) cnt += min(it.c, i);
    int j = cnt / i;
    if (j>=i && i*j>res) res = i*j, x = i, y = j;
  }

  cout << res << endl;
  cout << x << " " << y << endl;

  vector<int> vv;
  for (node it : vec) {
    int val = it.val, c = min(it.c, x);
    while ((int)vv.size()<res && c) vv.push_back(val), --c;
  }

  for (int i = 0; i <= x+5; ++i) w[i].resize(y+5);
  for (int id = 0, s_y = 1; s_y <= y; ++s_y) {
    for (int i = 1, j = s_y; i <= x; ++i, ++j) {
      if (j > y) j = 1;
      w[i][j] = vv[id];
      ++id;
    }
  }

  for (int i = 1; i <= x; ++i) {
    for (int j = 1; j <= y; ++j) cout << w[i][j] << " ";
    cout << endl;
  }
  return 0;
}
```

U477566 质因数分解

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

using namespace std;

void split(int x) {
    for (int i = 2; i <= x / i; ++i) {
        if (x % i == 0) {
            int cnt = 0;
            while (x % i == 0) { x /= i; ++cnt; }
            cout << i << " " << cnt << endl;
        }
    }
    if (x != 1) cout << x << " 1" << endl;
}
```

```
int main()
{
    int n; cin >> n;
    while (n -- ) {
        int x; cin >> x;
        split(x);
    }
    return 0;
}
```