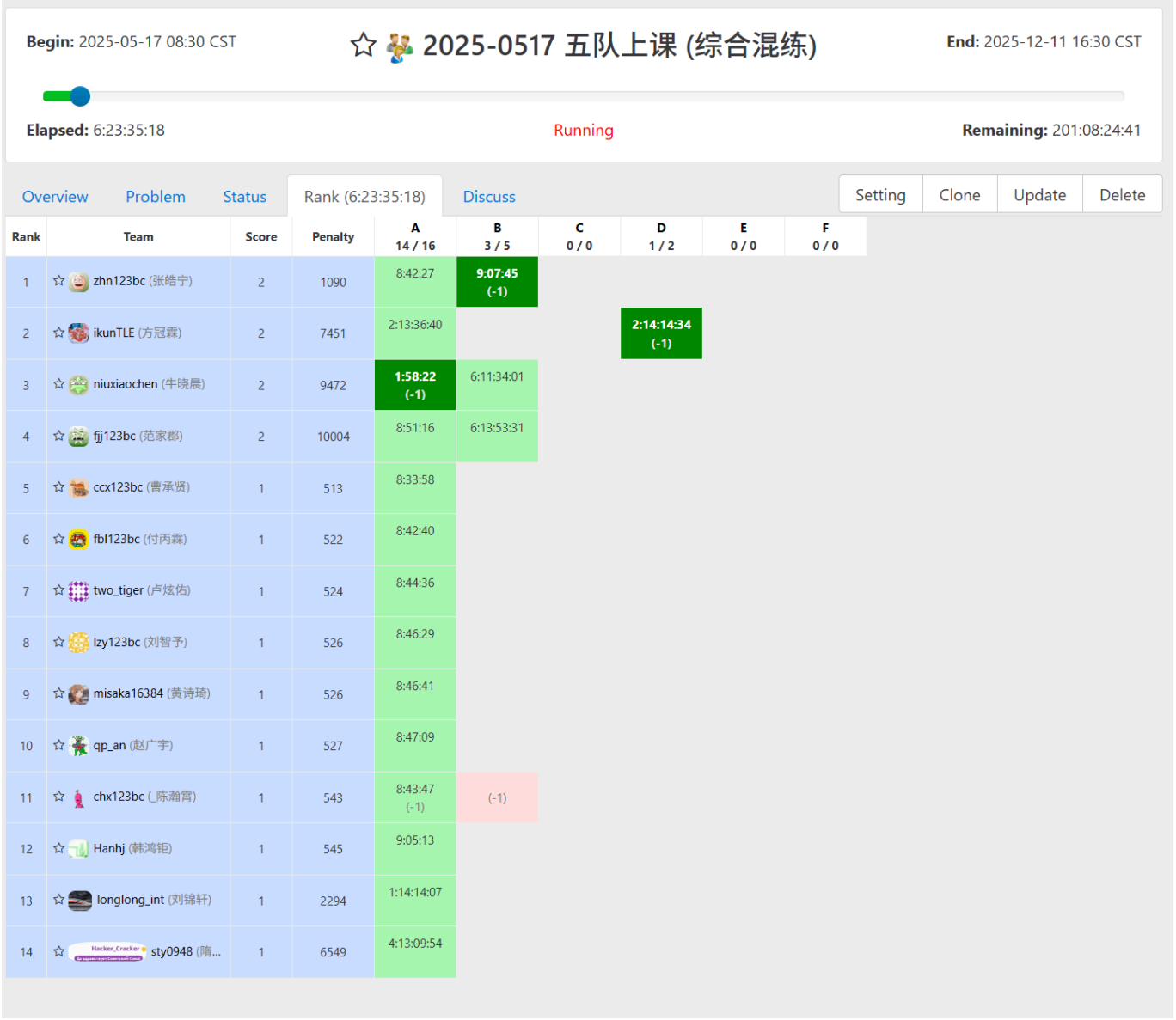
综合混练

人员

牛晓晨、刘新睿、隋天翼、范家郡 到课

上周作业检查

上周作业链接: https://vjudge.net/contest/717355



Begin: 2025-05-17 08:30 CST

☆ 2025-0517 五队上课 (综合混练)

End: 2025-12-11 16:30 CST

Elapsed: 6:23:35:18 Running Remaining: 201:08:24:41

Overview Problem Status Rank (6:23:35:18) Discuss Setting Clone Update Delete

Rank	Team	Score	Penalty	A 14 / 16	B 3 / 5	C 0 / 0	D 1 / 2	E 0 / 0	F 0 / 0
1	☆ zhn123bc (张皓宁)	2	1090	8:42:27	9:07:45 (-1)				
2	☆ ikunTLE (方冠霖)	2	7451	2:13:36:40			2:14:14:34 (-1)		
3	☆ niuxiaochen (牛晓晨)	2	9472	1:58:22 (-1)	6:11:34:01				
4	☆ fjj123bc (范家郡)	2	10004	8:51:16	6:13:53:31				
5	☆ ccx123bc (曹承贤)	1	513	8:33:58					
6	☆ fbl123bc (付丙霖)	1	522	8:42:40					
7	☆ two_tiger (卢炫佑)	1	524	8:44:36					
8	☆ lzy123bc (刘智予)	1	526	8:46:29					
9	☆ misaka16384 (黄诗琦)	1	526	8:46:41					
10	☆ qp_an (赵广宇)	1	527	8:47:09					
11	☆ chx123bc (_陈瀚霄)	1	543	8:43:47 (-1)	(-1)				
12	☆ Hanhj (韩鸿钜)	1	545	9:05:13					
13	☆ longlong_int (刘锦轩)	1	2294	1:14:14:07					
14	☆ sty0948 (隋...	1	6549	4:13:09:54					

作业

https://vjudge.net/contest/719073 (课上讲了上周比赛的 C D E F, 课后作业是本周比赛的 A B C D E 题)

课堂表现

同学们课上听讲很认真, 现在的问题的就是很多同学之前的题还欠着许多, 一定要花时间补之前的题。

课堂内容

CF981D Bookshelves

从 二进制高位 枚举到 二进制低位, 然后再用 dp 判断能否把这一位变成 1 即可

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

using namespace std;

typedef long long LL;
const int maxn = 50 + 5;
LL w[maxn], p[maxn];
bool f[maxn][maxn];
int n, m;

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

bool check(LL x) {
  memset(f, false, sizeof(f));
  f[0][0] = true;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      for (int k = 0; k <= i-1; ++k) {
        if (f[k][j-1] && (get_sum(k+1,i)&x)==x) f[i][j] = true;
      }
    }
  }
  return f[n][m];
}

int main()
{
  cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i], p[i] = p[i-1] + w[i];

  LL res = 0;
  for (int i = 60; i >= 0; --i) {
    if (check(res+(1LL<<i))) res += (1LL<<i);
  }
  cout << res << endl;
  return 0;
}
```

CF1198A MP3

每个数字最多用 m * 8 / n 位, 所以最多能表示 (1<<(m*8/n)) 个不同的数字

把所有数排序, 双指针扫一遍即可

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

```cpp
using namespace std;

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

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i];

  m = m * 8 / n;
  if (m >= 20) { cout << 0 << endl; return 0; }

  m = (1<<m);
  sort(w+1, w+n+1);

  int res = 0;
  map<int, int> mp;
  for (int l = 1, r = 1; r <= n; ++r) {
    mp[w[r]]++;
    while ((int)mp.size() > m) {
      mp[w[l]]--;
      if (mp[w[l]] == 0) mp.erase(w[l]);
      ++l;
    }
    res = max(res, r-l+1);
  }
  cout << n - res << endl;
  return 0;
}
```

CF366C Dima and Salad

要求 a/b == k, 则可以转化为 a - k*b == 0, 然后做 01 背包找最大值即可

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

using namespace std;

const int N = 10000 + 5;
int a[N], b[N], p[2*N], f[2*N];

int main()
{
  int n, k; cin >> n >> k;
  for (int i = 1; i <= n; ++i) cin >> a[i];
  for (int i = 1; i <= n; ++i) cin >> b[i];

  memset(p, -0x3f, sizeof(p)); p[N] = 0;
  memset(f, -0x3f, sizeof(f)); f[N] = 0;
  for (int i = 1; i <= n; ++i) {
```

```
    int x = a[i] - k*b[i];
    for (int j = 0; j < 2*N; j++) {
      if (j-x>=0 && j-x<2*N) f[j] = max(f[j], p[j-x] + a[i]);
    }
    memcpy(p, f, sizeof(p));
  }

  cout << (f[N]>0 ? f[N] : -1) << endl;
  return 0;
}
```

CF417D Cunning Gena

先按照 k 从小到大排序, 考虑以每个 k 进行结尾时, 做 状压dp 处理即可

```
#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, value;
  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, value = 0; cin >> x >> k >> m;
    while (m -- ) { int c; cin >> c; --c; value |= (1<<c); }
    w[i] = {x, k, value};
  }
  sort(w+1, w+n+1);

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

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