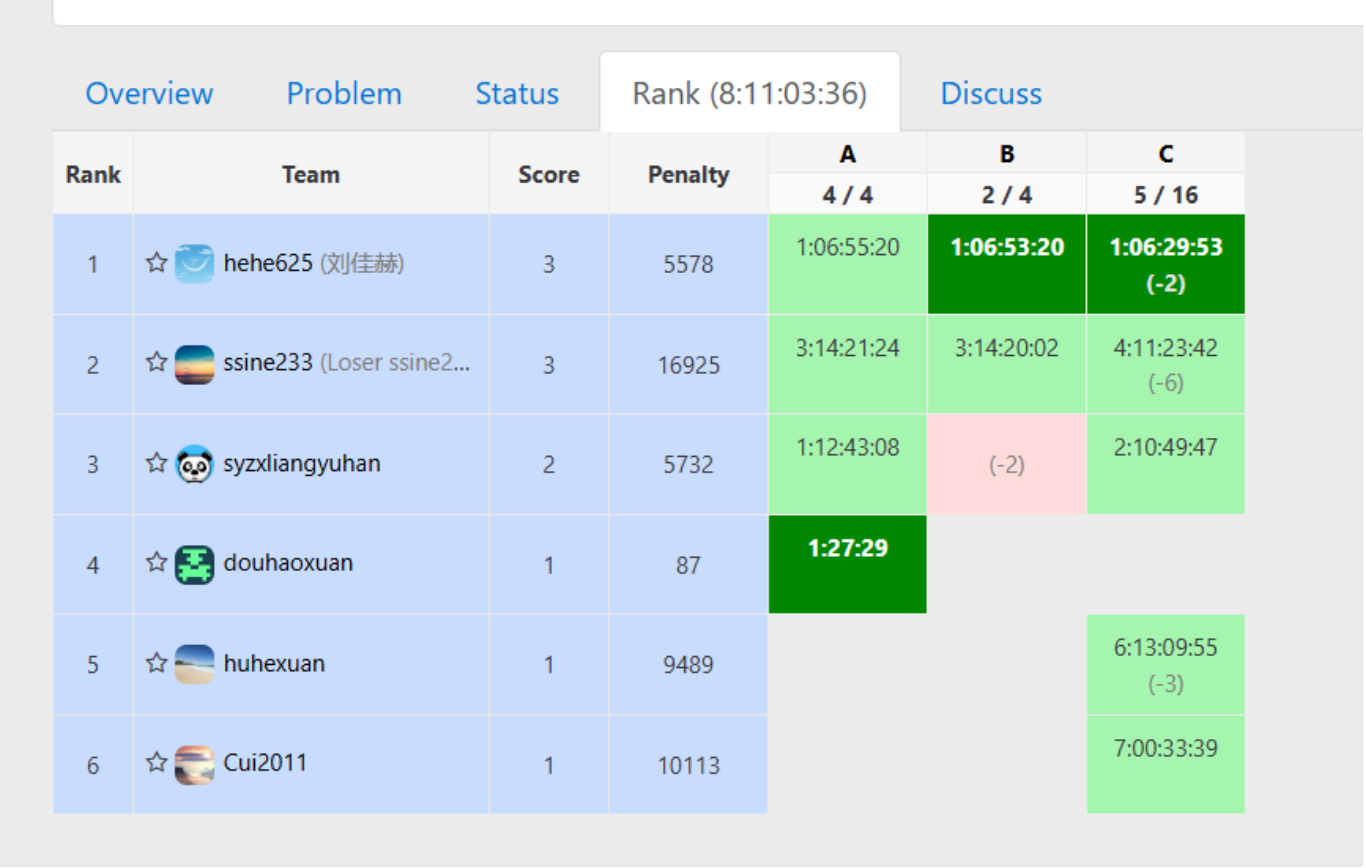
整除分块

人员

刘佳赫、左子毅、于珈浩、杨洋、朱奕鸣 到课

上周作业检查



Overview Problem Status Rank (8:11:03:36) Discuss

Rank	Team	Score	Penalty	A 4 / 4	B 2 / 4	C 5 / 16
1	☆ hehe625 (刘佳赫)	3	5578	1:06:55:20	**1:06:53:20**	**1:06:29:53** **(-2)**
2	☆ ssine233 (Loser ssine2...	3	16925	3:14:21:24	3:14:20:02	4:11:23:42 (-6)
3	☆ syzxliangyuhan	2	5732	1:12:43:08	(-2)	2:10:49:47
4	☆ douhaoxuan	1	87	**1:27:29**		
5	☆ huhexuan	1	9489			6:13:09:55 (-3)
6	☆ Cui2011	1	10113			7:00:33:39

作业

https://vjudge.net/contest/672619

课堂表现

今天课上做题时间比较少，主要以讲题目为主了，同学们课下一定要挤时间补题。

课堂内容

CF1594D The Number of Imposters

带权并查集 + 每个联通块单独处理

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

using namespace std;
```

```cpp
void print(int x) { cout << x << endl; }

const int maxn = 2e5 + 5;
const int mod = 2;
int f[maxn], d[maxn], h[maxn][2];
struct node {
  int op, a, b;
};

int fFind(int x) {
  if (f[x] != x) {
    int p = f[x];
    f[x] = fFind(f[x]);
    d[x] = (d[x] + d[p]) % mod;
  }
  return f[x];
}

void solve() {
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) f[i] = i, d[i] = 0, h[i][0] = h[i][1] = 0;

  vector<node> vec;
  while (m -- ) {
    int a, b; string op; cin >> a >> b >> op;
    if (op == "crewmate") vec.push_back({0, a, b});
    else vec.push_back({1, a, b});
  }

  for (node it : vec) {
    int op = it.op, a = it.a, b = it.b;
    int p = fFind(a), q = fFind(b);
    if (p==q && d[a]!=(d[b]+op)%mod) return print(-1);
    if (p != q) f[p] = q, d[p] = (d[b]+op - d[a] + mod) % mod;
  }

  for (int i = 1; i <= n; ++i) {
    int p = fFind(i); h[p][d[i]]++;
  }

  int res = 0;
  for (int i = 1; i <= n; ++i) res += max(h[i][0], h[i][1]);
  cout << res << endl;
}

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

CF459E Pashmak and Graph

按照边权分到每个桶, 然后按顺序遍历每个桶, 进行递推

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

using namespace std;

const int N = 3e5 + 5, M = 1e5 + 5;
struct node {
  int from, to;
};
vector<node> vec[M];
int f[N], p[N];

int main()
{
  int n, m; cin >> n >> m;
  while (m -- ) {
    int u, v, w; cin >> u >> v >> w;
    vec[w].push_back({u, v});
  }

  for (int i = 1; i < M; ++i) {
    for (node it : vec[i]) f[it.to] = max(f[it.to], p[it.from]+1);
    for (node it : vec[i]) p[it.to] = f[it.to];
  }

  int res = 0;
  for (int i = 1; i <= n; ++i) res = max(res, f[i]);
  cout << res << endl;
  return 0;
}
```

P2261 [CQOI2007] 余数求和

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

using namespace std;

typedef long long LL;

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

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

```
    if (t == 0) break;
    j = min(k/t, n);
    res -= get_sum(i, j) * t;
  }
  cout << res << endl;
  return 0;
}
```

CF148E Porcelain

可以将问题转化为 分组背包

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

using namespace std;

const int maxn = 100 + 5;
int f[maxn][maxn];  // f[i][j]: 第 i 组选 j 个时的最大值
int w[maxn], p[maxn];
int dp[maxn*maxn];  // dp[i]: 当只能选 i 个时，能取到的最大价值

int get_sum(int l, int r) { return (l<=r ? p[r]-p[l-1] : 0); }

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

  for (int len = 1; len <= n; ++len) {
    for (int i = 0; i <= len; ++i) {
      int j = n - (len-i) + 1;
      // 1~i, j~n
      f[id][len] = max(f[id][len], p[i]+get_sum(j,n));
    }
  }
}

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

  for (int i = 1; i <= n; ++i) {
    for (int j = m; j >= 1; --j) {
      for (int k = 1; k <= 100; ++k) {
        if (j < k) break;
        dp[j] = max(dp[j], dp[j-k] + f[i][k]);
      }
    }
  }
  cout << dp[m] << endl;
```

```
    return 0;
}
```