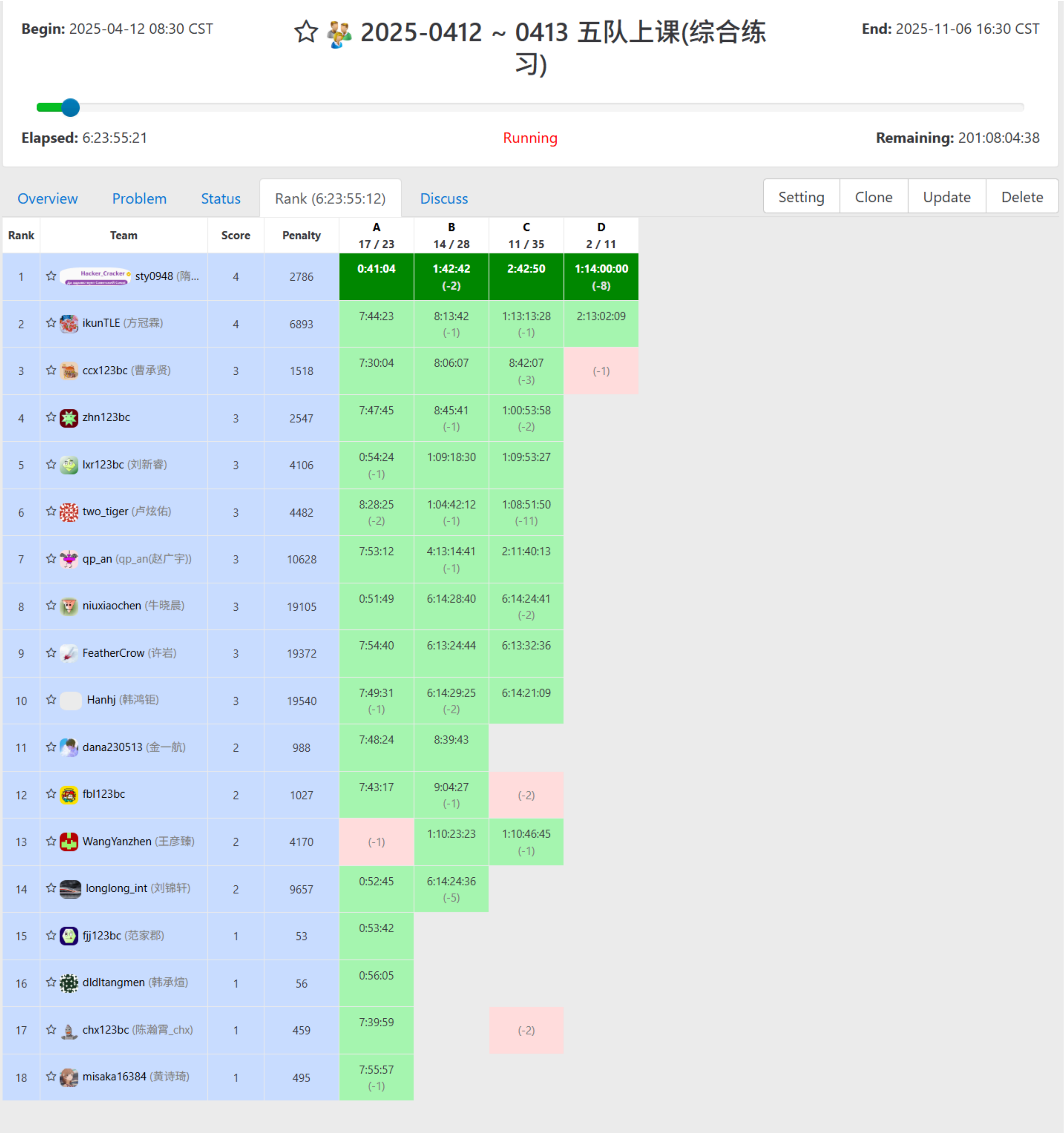
综合混练

人员

隋天翼、刘新睿、牛晓晨、韩承煊 到课

上周作业检查

上周作业链接: https://vjudge.net/contest/708790



Begin: 2025-04-12 08:30 CST

☆ 2025-0412 ~ 0413 五队上课(综合练习)

End: 2025-11-06 16:30 CST

Elapsed: 6:23:55:21

Running

Remaining: 201:08:04:38

Rank	Team	Score	Penalty	A 17 / 23	B 14 / 28	C 11 / 35	D 2 / 11
1	☆ sty0948 (隋...	4	2786	0:41:04	1:42:42 (-2)	2:42:50	1:14:00:00 (-8)
2	☆ ikunTLE (方冠霖)	4	6893	7:44:23	8:13:42 (-1)	1:13:13:28 (-1)	2:13:02:09
3	☆ ccx123bc (曹承贤)	3	1518	7:30:04	8:06:07	8:42:07 (-3)	(-1)
4	☆ zhn123bc	3	2547	7:47:45	8:45:41 (-1)	1:00:53:58 (-2)	
5	☆ lxr123bc (刘新睿)	3	4106	0:54:24 (-1)	1:09:18:30	1:09:53:27	
6	☆ two_tiger (卢炫佑)	3	4482	8:28:25 (-2)	1:04:42:12 (-1)	1:08:51:50 (-11)	
7	☆ qp_an (qp_an(赵广宇))	3	10628	7:53:12	4:13:14:41 (-1)	2:11:40:13	
8	☆ niuxiaochen (牛晓晨)	3	19105	0:51:49	6:14:28:40	6:14:24:41 (-2)	
9	☆ FeatherCrow (许岩)	3	19372	7:54:40	6:13:24:44	6:13:32:36	
10	☆ Hanhj (韩鸿钜)	3	19540	7:49:31 (-1)	6:14:29:25 (-2)	6:14:21:09	
11	☆ dana230513 (金一航)	2	988	7:48:24	8:39:43		
12	☆ fbl123bc	2	1027	7:43:17	9:04:27 (-1)	(-2)	
13	☆ WangYanzhen (王彦臻)	2	4170	(-1)	1:10:23:23	1:10:46:45 (-1)	
14	☆ longlong_int (刘锦轩)	2	9657	0:52:45	6:14:24:36 (-5)		
15	☆ fjj123bc (范家郡)	1	53	0:53:42			
16	☆ dldltangmen (韩承煊)	1	56	0:56:05			
17	☆ chx123bc (陈瀚霄_chx)	1	459	7:39:59		(-2)	
18	☆ misaka16384 (黄诗琦)	1	495	7:55:57 (-1)			

作业

https://vjudge.net/contest/710557 (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

今天同学们上课听讲都很认真, 不过做题时表现不太好, 体现在调 bug 很吃力。

老师专门给同学们讲了一些如何调 bug, 同学们以后调试也要先思考如何加输出, 不要蒙着头调试。

课堂内容

CF1901D Yet Another Monster Fight

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

using namespace std;

const int maxn = 3e5 + 5;
int w[maxn], pre_maxx[maxn], suf_maxx[maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) {
    cin >> w[i]; pre_maxx[i] = max(pre_maxx[i-1], w[i] + n-i);
  }
  for (int i = n; i >= 1; --i) suf_maxx[i] = max(suf_maxx[i+1], w[i] + i-1);

  int res = 2e9;
  for (int i = 1; i <= n; ++i) res = min(res, max({w[i], pre_maxx[i-1], suf_maxx[i+1]}));
  cout << res << endl;
  return 0;
}
```

P1106 删数问题

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

using namespace std;

int main()
{
  string s; cin >> s;
  int k; cin >> k;

  vector<char> vec;
  for (char i : s) {
    while (!vec.empty() && i<vec.back() && k) vec.pop_back(), --k;
    vec.push_back(i);
  }

  while (k) vec.pop_back(), --k;

  reverse(vec.begin(), vec.end());
  while ((int)vec.size()>=2 && vec.back()=='0') vec.pop_back();
  reverse(vec.begin(), vec.end());

  for (char i : vec) cout << i;
  cout << endl;
  return 0;
}
```

CF1896D Ones and Twos

```
// 方法一: 二分 + 树状数组
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
int w[maxn], tr[maxn];

int lowbit(int x) { return x&(-x); }
void update(int x, int k) {
  while (x < maxn) tr[x] += k, x += lowbit(x);
}
int query(int x) {
  int res = 0;
  while (x) res += tr[x], x -= lowbit(x);
  return res;
}

int n, sum;

int queryL() {
  int l = 1, r = n;
  while (l <= r) {
    int mid = (l + r) / 2;
    if (query(mid) < mid*2) r = mid-1;
    else l = mid+1;
  }

  return (l==0 ? -1 : l);
}
int queryR() {
  int l = 1, r = n;
  while (l <= r) {
    int mid = (l + r) / 2;
    if (sum-query(mid-1) < (n-mid+1)*2) l = mid+1;
    else r = mid-1;
  }

  return (r==n+1 ? -1 : r);
}

void solve() {
  int m; cin >> n >> m;
  sum = 0;
  for (int i = 1; i <= n; ++i) cin >> w[i], sum += w[i], update(i, w[i]);

  while (m -- ) {
    int op; cin >> op;
    if (op == 1) {
      int value; cin >> value;
      if (value > sum) cout << "NO" << endl;
      else if ((sum-value) % 2 == 0) cout << "YES" << endl;
      else {
        int l = queryL(), r = queryR();
        if (l==-1 && r==-1) cout << "NO" << endl;
        else {
          int t;
          if (l == -1) t = n-r+1;
          else if (r == -1) t = l;
          else t = min(l, n-r+1);

          if (value <= sum-(2*t-1)) cout << "YES" << endl;
          else cout << "NO" << endl;
        }
      }
    } else {
      int pos, value; cin >> pos >> value;
      sum -= w[pos]; update(pos, -w[pos]);
      w[pos] = value; sum += w[pos]; update(pos, w[pos]);
    }
  }

  for (int i = 1; i <= n; ++i) update(i, -w[i]), w[i] = 0;
}

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

```
// 方法二: set 维护 1 的位置
#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
```

```
int w[maxn];

set<int> s;

int n, sum;

int queryL() {
  if (s.empty()) return -1;
  return *s.begin();
}
int queryR() {
  if (s.empty()) return -1;
  return *s.rbegin();
}

void solve() {
  s.clear();

  int m; cin >> n >> m;
  sum = 0;
  for (int i = 1; i <= n; ++i) {
    cin >> w[i], sum += w[i];
    if (w[i] == 1) s.insert(i);
  }

  while (m -- ) {
    int op; cin >> op;
    if (op == 1) {
      int value; cin >> value;
      if (value > sum) cout << "NO" << endl;
      else if ((sum-value) % 2 == 0) cout << "YES" << endl;
      else {
        int l = queryL(), r = queryR();
        if (l==-1 && r==-1) cout << "NO" << endl;
        else {
          int t;
          if (l == -1) t = n-r+1;
          else if (r == -1) t = l;
          else t = min(l, n-r+1);

          if (value <= sum-(2*t-1)) cout << "YES" << endl;
          else cout << "NO" << endl;
        }
      }
    } else {
      int pos, value; cin >> pos >> value;
      if (w[pos] == 1) s.erase(pos);
      sum -= w[pos]; w[pos] = value; sum += w[pos];
      if (w[pos] == 1) s.insert(pos);
    }
  }
}

int main()
```

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

CF7D Palindrome Degree

```
#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], h[maxn], h2[maxn];
int f[maxn];

ULL get_hash(int l, int r) { return h[r] - h[l-1]*p[r-l+1]; }
ULL get_hash_2(int l, int r) { return h2[l] - h2[r+1]*p[r-l+1]; }

int main()
{
  scanf("%s", s+1);
  int n = strlen(s+1);

  p[0] = h[0] = 1;
  for (int i = 1; i <= n; ++i) {
    p[i] = p[i-1]*P; h[i] = h[i-1]*P + s[i];
  }
  h2[n+1] = 1;
  for (int i = n; i >= 1; --i) h2[i] = h2[i+1]*P + s[i];

  LL res = 0;
  for (int i = 1; i <= n; ++i) {
    if (get_hash(1,i) == get_hash_2(1,i)) f[i] = f[i/2] + 1;
    res += f[i];
  }
  cout << res << endl;
  return 0;
}
```