

2023 牛客暑期多校训练营1

整体过题记录 AC **(4/13)

A	B	C	D	E	F	G	H	I	J	K	L	M
4/288			1/28		-1		1/135		1/56	-3		

比赛链接: <https://ac.nowcoder.com/acm/contest/57355>

题目分析及错误反思

A. Almost Correct

题型: 构造题

罚时原因: 小蔡给黄陈传递的题意出错。

H. Matches

题型: 传统题

可以分析出只有交换 $a_i > b_i$ 和 $a_j < b_j$ 的两组的时候才可以让答案变得更小。

所以可以直接按照 r 排序, 然后倒着扫一遍, 更新 l , 就可以得到覆盖最大的长度了。

答案为 $ans - 2 * len$, ans 为一开始的答案, len 为最长的覆盖长度。

K. Subdivision

题型: HC: 思维 伪拓扑 贪心

赛中没有过, 但是和其他队伍交流后知道做法没有问题, 所以是代码实现的问题。

赛后补题安排

题目分配

- H: K,
- C: K
- Z: M、L

题目记录

K

```
#include <bits/stdc++.h>
#define N 200005
using namespace std;
int dis[N], vis[N], n, m, k;
```

```

long long ans;
vector<int> e[N], edge[N];
map<pair<int, int>, int> mp;
void bfs() {
    queue<int> q;
    for (int i = 1; i <= n; i++) dis[i] = 1e9 + 5;
    q.push(1);
    dis[1] = 0;
    vis[1] = 1;
    while (!q.empty()) {
        int x = q.front();
        // cout << x << endl;
        q.pop();
        for (int i = 0; i < e[x].size(); i++) {
            int y = e[x][i];
            // cout << y << endl;
            if (vis[y] == 0) {
                vis[y] = 1;
                q.push(y);
                edge[x].push_back(y);
                mp[{x, y}] = 1;
                mp[{y, x}] = 1;
                // cout << y << endl;
            }
            dis[y] = min(dis[y], dis[x] + 1);
        }
    }
}

void dfs(int x) {
    // cout << x << endl;
    if (dis[x] <= k)
        ans++;
    for (auto y : edge[x]) {
        for (auto u : e[y])
            if (mp[{u, y}] == 0) {
                ans += 1ll * max(0, (k - dis[y]));
            }
        if (edge[y].size() == 0) {
            int tot = 0;
            for (int u : e[y])
                if (mp[{u, y}] == 0)
                    tot++;
            if (tot == 0)
                ans += 1ll * max(0, k - dis[y]);
        }
        dfs(y);
    }
}

int main() {
    scanf("%d%d%d", &n, &m, &k);
    for (int i = 1; i <= m; i++) {
        int x, y;
        cin >> x >> y;
        e[x].push_back(y);
        e[y].push_back(x);
    }
}

```

```

    }
    bfs();
    dfs(1);
    printf("%lld\n", ans);
}

```

M. Water

就是每一个 $Ax + By$ 都是可以通过若干操作倒出来，所以可以用 exgcd 求解出答案。

exgcd 求解出的可行的解，因为 x, y 的正负都有可能。

- $x > 0, y > 0$ 时, $ans = 2(x + y)$ 。
- $xy < 0$ 时, $ans = 2|x - y| - 1$ 。

```

#include <bits/stdc++.h>

using namespace std;

using i64 = long long;

i64 exgcd(i64 a, i64 b, i64& x, i64& y) {
    if (b == 0) {
        x = 1, y = 0;
        return a;
    }
    i64 g = exgcd(b, a % b, y, x);
    y -= (a / b) * x;
    return g;
}

void solve() {
    i64 A, B, s;
    cin >> A >> B >> s;
    if (A < B) {
        swap(A, B);
    }
    i64 x, y;
    i64 g = exgcd(A, B, x, y);
    if (s % g != 0) {
        cout << -1 << "\n";
        return;
    }
    i64 ans = 1e18;

    A /= g, B /= g, s /= g;
    i64 x = (x * s % B + B) % B;
    i64 y = (s - A * x) / B;
    for (i64 i = -1; i <= 1; i++) {
        i64 xx = x - i * B, yy = y + i * A;
        if (xx * yy < 0) {
            ans = min(ans, 2ll * abs(xx - yy) - 1);
        } else {
            ans = min(ans, 2 * (xx + yy));
        }
    }
}

```

```

    }
    y = (Y * s % A + A) % A;
    x = (s - B * y) / A;
    for (i64 i = -1; i <= 1; i++) {
        i64 xx = x - i * B, yy = y + i * A;
        if (xx * yy < 0) {
            ans = min(ans, 211 * abs(xx - yy) - 1);
        } else {
            ans = min(ans, 2 * (xx + yy));
        }
    }
    cout << ans << "\n";
}

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

```

暴露问题及需要补的知识点

暴露的问题

- 因为队伍采取的策略是前中期的题目尽量个人完成，所以会出现如果一个人在某道题目卡住的时候，会出现有个人需要 debug 的问题，影响队伍出题的稳定性，前中期尽量还是需要提高自己的代码稳定性。

需要补的知识点

一定要解决：这场的数学题，小张表现得非常不好，需要加油。