

思考题讲解2+求逆元

人员

蔡云翔、石宇赫、李佳声、胡赫轩、崔嘉睿、穆鹏宇、程晟泰、梁钰涵、周子航 到课

作业检查

于潇涵 完成了一部分

王梓同 上周请假

蔡云翔 上周请假

石宇赫 已完成

李佳声 上周请假

胡赫轩 完成了一部分

崔嘉睿 已完成

穆鹏宇 未完成

程晟泰 完成了一部分

梁钰涵 完成了一部分

周子航 已完成

作业

<https://www.luogu.com.cn/contest/180108>, A ~ F 题要求同学们补完。H、I、J 三个题作为课下思考题。

课堂表现

同学们听课都很认真，不过课上讲的题目有些多，同学们一定要课下好好复习，只凭课上时间是不能完全消化理解的。

课堂内容

T464555 大鱼吃小鱼

可以二分做，也可以双指针做

一定是大鱼可以全吃完，小鱼不可以

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

using namespace std;
```

```

typedef long long LL;
const int maxn = 5e5 + 5;
int a[maxn], b[maxn];

bool check(int n, int mid) {
    LL sum = b[mid];
    for (int i = 1; i <= n; ++i) {
        if (i == mid) continue;
        if (b[i] < sum) sum += b[i];
        else return false;
    }
    return true;
}

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

    memcpy(b, a, sizeof(b));
    sort(b+1, b+n+1);
    int l = 1, r = n;
    while (l <= r) {
        int mid = (l+r) / 2;
        if (check(n, mid)) r = mid-1;
        else l = mid+1;
    }

    for (int i = 1; i <= n; ++i) {
        if (l<=n && a[i]>=b[l]) cout << "T";
        else cout << "N";
    }
    cout << endl;
    return 0;
}

```

T459972 Wandering

前缀和，同时维护一个前缀最大值的数组

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn];
LL p[maxn], pMax[maxn];

int main()

```

```

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

    LL pos = 0, res = 0;
    for (int i = 1; i <= n; ++i) {
        res = max(res, pos + pMax[i]);
        pos += p[i];
    }
    cout << res << endl;
    return 0;
}

```

T466240 playing

线性dp

$f[i][j][k][0/1]$: 代表对于第 i 个数来说, $i-2 \sim i$ 一起减 j 次, $i-3 \sim i-1$ 一起减 k 次, $i-2$ 的前面是否有对子

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 100 + 5;
bool f[maxn][3][3][2], st[maxn];
int w[maxn];

void solve() {
    memset(f, false, sizeof(f));
    for (int i = 1; i <= 100; ++i) cin >> w[i];

    f[1][0][0][0] = f[2][0][0][0] = true;
    for (int i = 3; i <= 102; ++i) {
        for (int j = 0; j < 3; ++j) { // i
            for (int k = 0; k < 3; ++k) { // i-1
                for (int l = 0; l < 3; ++l) { // i-2
                    if (j > w[i] || j+k > w[i-1] || j+k+l > w[i-2]) continue;
                    for (int t = 0; t < 2; ++t) f[i][j][k][t] |= (f[i-1][k][l][t] && st[w[i-2]-(j+k+l)]);
                    if (j+k+l+2 <= w[i-2] && st[w[i-2]-j-k-l-2]) f[i][j][k][1] |= f[i-1][k][l][1];
                }
            }
        }
    }

    cout << (f[102][0][0][1] ? "Yes" : "No") << endl;
}

```

```

}

int main()
{
    for (int i = 0; i < maxn; ++i) st[i] = true;
    st[1] = st[2] = st[5] = false;

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

```

T460084 flip

思维题，需要特判几种特殊情况

```

#include <bits/stdc++.h>

using namespace std;

void solve() {
    int n; cin >> n;
    string str; cin >> str;
    int cnt = 0;
    for (char i : str) {
        if (i == '1') ++cnt;
    }

    if (cnt&1) { cout << -1 << endl; return; }
    if (cnt==0 || cnt>=4) { cout << cnt/2 << endl; return; }

    if (str=="011" || str=="110") { cout << -1 << endl; return; }
    if (str=="0110") { cout << 3 << endl; return; }
    for (int i = 0; i < (int)str.size()-1; ++i) {
        if (str[i]=='1' && str[i+1]=='1') { cout << 2 << endl; return; }
    }
    cout << 1 << endl;
}

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

```

T460085 place

双指针，总后往前考虑

```

#include <bits/stdc++.h>

using namespace std;

bool match(string s, string t) {
    sort(s.begin(), s.end());
    sort(t.begin(), t.end());
    return s==t;
}

int main()
{
    int n; cin >> n;
    string s, t; cin >> s >> t;
    if (!match(s, t)) { cout << -1 << endl; return 0; }

    for (int i=n-1, j=n-1; j>=0; --j) {
        if (s[i] == t[j]) --i;
        if (!j) cout << i+1 << endl;
    }
    return 0;
}

```

T465967 add

用桶记录每个数出现的位置，前缀和维护每个位置分别有多少数

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
LL value[maxn], p[maxn];
bool st[maxn];
vector<int> vec[maxn];

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

int main()
{
    int n, m, cnt = 0; cin >> n >> m;
    for (int i = 1; i <= m; ++i) {
        int x; cin >> x;
        st[x] = !st[x];

        if (st[x]) ++cnt;
        else --cnt;

        p[i] = cnt;
    }
}

```

```

    vec[x].push_back(i);
}

for (int i = 1; i <= m; ++i) p[i] += p[i-1];

for (int i = 1; i <= n; ++i) {
    int len = vec[i].size();
    if (len & 1) { vec[i].push_back(m+1); ++len; }

    LL res = 0;
    for (int j = 0; j < len; j += 2) res += sum(vec[i][j], vec[i][j+1]-1);
    cout << res << " ";
}
cout << endl;
return 0;
}

```

求逆元模板 (只能在模数是质数的时候用,一般95%以上的题目模数都是998244353或者 $1e9+7$,都可以用这种方法)

```

typedef long long LL;
const int mod = 998244353;

int qmod(int a, int k) {
    int res = 1;
    while (k) {
        if (k&1) res = (LL)res*a % mod;
        a = (LL)a*a % mod;
        k >>= 1;
    }
    return res;
}

int inv(int x) { return qmod(x, mod-2); }

```

期望

如果一个数成为 1 的概率是 p_1 , 成为 2 的概率是 p_2 , ..., 成为 n 的概率是 p_n

那么它的最终期望是 $1*p_1 + 2*p_2 + \dots + n*p_n$