2021-11-14周赛(第一场)

T1

1. 题意

K个月之后是啥月

2. 思路

直接做就行

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>
#include <map>

using namespace std;

#define ll long long

typedef pair<int, int> pii;
typedef pair<ll, ll> pll;

string str;
int K;
map<string, int> mp1;
map<int, string> mp2;

void init() {
    mp1["January"] = 0;
    mp1["February"] = 1;
    mp1["March"] = 2;
    mp1["April"] = 3;
    mp1["May"] = 4;
    mp1["June"] = 5;
    mp1["July"] = 6;
    mp1["August"] = 7;
    mp1["September"] = 8;
    mp1["October"] = 9;
    mp1["November"] = 10;
    mp1["December"] = 11;

    for (auto [a, b] : mp1) {
        mp2[b] = a;
    }
}

void solve() {
    init();

    int cur = mp1[str];
```

```
    cur = (cur + K) % 12;
    cout << mp2[cur] << endl;
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> str >> K;
    solve();

    return 0;
}
```

T2

1. 题意

l到r之间, k的次幂有哪几个

2. 思路

因为k >= 2, 所以最差情况下是O(log2(n))的, 直接求出上界, 一步一步算就行.

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>
#include <cmath>

using namespace std;

#define ll long long

typedef pair<int, int> pii;
typedef pair<ll, ll> pll;

ll L, R, K;

void solve() {
    ll lim = floor(log(R) / log(K));
    ll cur = 1, cnt = 0, flag = 0;
    while (cnt++ <= lim) {
        if (cur >= L) {
            flag = 1;
            cout << cur << " ";
        }
        cur *= K;
    }
    if (!flag) cout << -1 << endl;
    cout << endl;
}
```

```cpp
int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> L >> R >> K;
    solve();

    return 0;
}
```

T3

1. 题意

给偏序, 求全序

2. 思路

人名是字符串, 做一下映射就行, 之后直接用拓扑求.

ps: 左边界必须先进入队列.

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>
#include <queue>
#include <unordered_map>

using namespace std;

#define ll long long

typedef pair<int, int> pii;
typedef pair<ll, ll> pll;

const int N = 1e4 + 10;

int n;
```

```cpp
int tt;
string mp1[N];
unordered_map<string, int> mp2;

int du[N];
vector<int> g[N];

void add(int u, int v) {
    g[u].push_back(v);
    du[v]++;
}

int get_id(string x) {
    if (!mp2.count(x)) {
        mp1[++tt] = x;
        mp2[x] = tt;
    }

    return mp2[x];
}

void solve() {
    vector<string> ans;
    queue<int> que;
    for (int i = 1; i <= tt; i++) {
        if (mp1[i] == "@") {
            que.push(i);
            break;
        }
    }
    for (int i = 1; i <= tt; i++) {
        if (du[i] == 0 && mp1[i] != "@") que.push(i);
    }

    while (!que.empty()) {
        int u = que.front(); que.pop();
        ans.push_back(mp1[u]);

        for (auto v : g[u]) {
            if (--du[v] == 0) que.push(v);
        }
    }
    for (string &x : ans) {
        if (x == "@" || x == "#") continue;
        cout << x << " ";
    }
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> n;
    while (n--) {
        string a, b;
```

```
        cin >> a >> b;
        if (a == "#") a = "@";

        int id1 = get_id(a);
        int id2 = get_id(b);
        add(id1, id2);
    }
    solve();

    return 0;
}
```

T4

1. 题意

给个大整数, 判断是否为质数

2. 思路

米勒-拉宾素性检验模板题

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>

using namespace std;

#define ll long long

typedef pair<int, int> pii;
typedef pair<ll, ll> pll;

ll n;
int times;

// 快速乘
ll qsc(ll a, ll b, ll m) {
    ll ans = 0;
    a %= m;
    while (b) {
        if (b & 1) {
            ans = (ans + a) % m;
            b--;
        }
        b >>= 1;
        a = (a + a) % m;
    }
    return ans;
}

// 快速幂
ll qsm(ll a, ll b, ll m) {
    ll ans = 1;
```

```
    a %= m;
    while (b) {
        if (b & 1) {
            ans = qsc(ans, a, m);
            b--;
        }
        b >>= 1;
        a = qsc(a, a, m);
    }
    return ans;
}

bool Miller_Rabin() {
    if (n == 2) return true;
    if (n < 2 || !(n & 1)) return false;
    ll m = n - 1;
    int k = 0;
    while ((m & 1) == 0) {
        k++;  // 这个k就是我们讲的时候的t
        m >>= 1;  // 这个m就是k
    }
    // Times就是我们事先定义的s（要找a的个数）
    for (int i = 0; i < times; i++) {
        ll a = rand() % (n - 1) + 1;  // 找一个[1,n-1]内的任意数
        ll x = qsm(a, m, n);
        ll y = 0;
        for (int j = 0; j < k; j++) {
            y = qsc(x, x, n);
            if (y == 1 && x != 1 && x != n - 1) return false;
            x = y;
        }
        if (y != 1) return false;
    }
    return true;
}

void solve() {
    times = 10;
    if (Miller_Rabin()) cout << "Yes" << endl;
    else cout << "No" << endl;
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    int T;
    cin >> T;
    while (T--) {
        cin >> n;
        solve();
    }

    return 0;
}
```

T5

1. 题意

对于每个询问, 求l到r之间两数异或最大值

2. 思路

n跟m较小, 对于每次询问, 直接建立一个trie树暴力求即可.
trie树求两数异或最大值不会的同学, 可以参考leetcode 421.

ps: 数字最大才1024, 所以不需要求到30位, 否则会tle.

以上是菜鸡做法, 复杂度有点极限, 很幸运, 卡过去了, 正解是区间dp.

f(i, j)表示区间[i, j]的异或最大值

f(i, j) = max{f(i, j - 1), f(i + 1, j), a_i ^ a_j}

3. 参考代码

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>

using namespace std;

#define ll long long

typedef pair<int, int> pii;
typedef pair<ll, ll> pll;

const int N = 5e3 + 10;

int n, m;
int va[11 * N];
int tree[11 * N][2], vis[11 * N], tt;

void insert(int x) {
    int rt = 0;
    for (int i = 11; i >= 0; i--) {
        int cur = (x >> i) & 1;
        if (!tree[rt][cur]) tree[rt][cur] = ++tt;
        rt = tree[rt][cur];
    }
    vis[rt] = 1;
}

int search(int x) {
    int ans = 0;
    int rt = 0;
    for (int i = 11; i >= 0; i--) {
        int cur = (x >> i) & 1;
        int tar = cur ^ 1;
```

```
        if (tree[rt][tar]) {
            ans |= tar << i;
            rt = tree[rt][tar];
        } else {
            ans |= cur << i;
            rt = tree[rt][cur];
        }
    }
    return ans;
}

int findMaximumXOR(vector<int>& nums) {
    int ans = 0;
    for (int x: nums) {
        insert(x);
        int y = search(x);
        ans = max(ans, x ^ y);
    }

    return ans;
}

void init() {
    memset(tree, 0, sizeof tree);
    memset(vis, 0, sizeof vis);
    tt = 0;
}

void solve() {
    while (m--) {
        init();
        int L, R;
        cin >> L >> R;
        vector<int> nums;
        for (int i = L; i <= R; i++) nums.push_back(va[i]);
        cout << findMaximumXOR(nums) << endl;
    }
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> n >> m;
    for (int i = 1; i <= n; i++) cin >> va[i];
    solve();

    return 0;
}
```

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
```

```cpp
#include <vector>

using namespace std;

#define ll long long

typedef pair<int, int> pii;
typedef pair<ll, ll> pll;

const int N = 5e3 + 10;

int n, m;
ll va[N];
ll f[N][N];

void solve() {
    for (int len = 2; len <= n; len++) {
        for (int L = 1; L + len - 1 <= n; L++) {
            int R = L + len - 1;
            f[L][R] = max({f[L + 1][R], f[L][R - 1], va[L] ^ va[R]});
        }
    }

    while (m--) {
        int L, R;
        cin >> L >> R;
        cout << f[L][R] << endl;
    }
}

int main() {
#ifdef LOCAL
    freopen("../in.txt", "r", stdin);
    freopen("../out.txt", "w", stdout);
#endif

    ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    cin >> n >> m;
    for (int i = 1; i <= n; i++) cin >> va[i];
    solve();

    return 0;
}
```

T6

计算器