

彰化高中資訊學科能力競賽 114 學年度 校內初選 題解

tags: HARC CHSH nhspc 題解 chsh-exam

A. 進位 *Carry*

出題者

Roy

首殺

楊順宇

標籤

sort

pair

for

進位轉換可以先寫這題 [ZJ a034](#)

或參考進位轉換教學 [網路亂找的講義](#)

排序看到很多人自己重載運算子，實際上用 `pair` 直接 `sort` 也可以喔～

task: 100% **As statement**

用了比較多近期競賽大家常用的語法，也可以學一下。看不懂Discord請踴躍提問～

```
#include <bits/stdc++.h>
using namespace std;
#define int long long

signed main() {
    int t;
    cin >> t;
    auto convert = [&](int x, int base) -> string {
        string res = "";
        while(x) {
            int a = x % base;
            x /= base;
            if(a < 10) res += char('0' + a);
            else res += char('A' + a - 10);
        }
        return string(res.rbegin(), res.rend());
    };
    while(t--) {
        int n;
        cin >> n;
        vector<pair<string, int>> ans;
        for(int i = 2; i <= 16; i++) {
            ans.emplace_back(convert(n, i), i);
        }
        sort(ans.begin(), ans.end());
    }
}
```

```

        for(auto [a, b] : ans) {
            cout << a << ' ' << b << endl;
        }
    }
}

```

B. 合成函數 *Composite Function*

出題者

MelonWalker

首殺

黃振哲

標籤

快速冪

subtask1: 30% $n \leq 10^6$

直接迭代即可，需注意的是合成函數其大小為指數性成長，所以如果最後才 mod 的話會溢位。
mod 對加法與乘法有分配律故每次迭代函數直接 mod 不影響答案 (注意除法不成立)

```

#include <bits/stdc++.h>
using namespace std;
using ll = long long;

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    ll n, m, a, b, x;
    cin >> n >> m >> a >> b >> x;

    ll ca = 1, cb = 0;
    for (int i = 0; i < n; i++) {
        tie(ca, cb) = make_pair((a * ca) % m, (a * cb + b) % m);
    }
    cout << (ca * x + cb) % m << '\n';
}

```

subtask2: 70% *As statement*

由於合成函數有此特性 $f^{2n} = f^n(f^n)$ ，所以事實上我們不用一步一步迭代。
可以發現這題解法跟快速冪很像，所以只要用 ~~operator overloading~~ 就能無腦套模板子。

```

#include <bits/stdc++.h>
using namespace std;

```

```

using ll = long long;

int m;

struct F {
    ll a, b;
    F operator*(F rh) {
        return {a * rh.a % m, (a * rh.b + b) % m};
    }
};

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    ll n, x;
    F f;
    cin >> n >> m >> f.a >> f.b >> x;

    F ans{1, 0};
    while (n) {
        if (n & 1) {
            ans = ans * f;
        }
        f = f * f;
        n >>= 1;
    }
    cout << (ans.a * x + ans.b) % m << '\n';
}

```

C. K-字串分割 *K-Limited String Partition*

出題者
Gemini

首殺
吳品諒

標籤
greedy

task: 100% *As statement*

不難發現，提早分割對於減少之後的分割無任何的幫助，所以對於每次分割當然字串越長越好。這裡把證明留給大家練習，其實是我不會。

```

#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(0);

```

```

cin.tie(0);

int n, k;
cin >> n >> k;

string s;
cin >> s;
int cnt[26] = {}, ans = 1;
for (char c: s) {
    if (cnt[c - 'a'] >= k) {
        memset(cnt, 0, sizeof(cnt));
        ans++;
    }
    cnt[c - 'a']++;
}
cout << ans << '\n';
}

```

D. AI 程式助理 *MelonGPT*

出題者

MelonWalker

首殺

黃振哲

標籤

輸入輸出

task: 100% **As statement**

會用 getline 應該都能寫出來。

```

#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    string s;
    getline(cin, s);
    if (s == "Why can't I run my code?") {
        cout << "Haven't you considered looking at your own problems first?\n";
    } else if (s == "What does this error message mean?") {
        cout << "This error message alienates the effort people put in, you get it?\n";
    } else if (s == "How can I optimize the performance of this code?") {
        cout << "Why bother being so concerned about performance?\n";
    } else {
        cout << "The server is busy. Please try again later.\n";
    }
}

```

```
}  
}
```

E. 貨物中心 *Pchome*

出題者

Roy

首殺

楊順宇

標籤

二維資料結構

task: 100% *As statement*

```
#include <bits/stdc++.h>  
using namespace std;  
  
const int MAXN = 1e5+5;  
int main() {  
    ios::sync_with_stdio(0);  
    cin.tie(0);  
  
    int q;  
    cin >> q;  
    vector<vector<string>> shelve(MAXN); // 1) 2D vector  
    // vector<string> shelve[MAXN];      // 2) 也可以用 vector + array  
  
    while (q--) {  
        int t, x;  
        cin >> t >> x;  
        x--;  
        if (t == 1) {  
            string s;  
            cin >> s;  
            shelve[x].push_back(s);  
        } else {  
            int y;  
            cin >> y;  
            y--;  
            if (y >= shelve[x].size()) {  
                cout << "oh can't find anything...\n";  
            } else {  
                cout << shelve[x][y] << '\n';  
            }  
        }  
    }  
}
```

task: 100% **As statement**

F. 蝸蝸牛牛 *Snails*

出題者

Roy

首殺

黃振哲

標籤

DFS

枚舉

task: 100% **As statement**

暴力枚舉所有可能路徑，再暴力比較就行了。

AC code 1

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int x1, y1;
    int x2, y2;
    cin >> x1 >> y1;
    cin >> x2 >> y2;
    string s1, s2;
    cin >> s1;
    cin >> s2;

    int dx[4] = {1, 0, -1, 0};
    int dy[4] = {0, -1, 0, 1};
    auto dfs = [&](auto sf, string s, int pos, int x, int y, vector<pair<int,
int>> v, vector<vector<pair<int,int>>> &p) -> void {

        if(pos == s.length()) {
            p.push_back(v);
            return;
        }
        int k;
        if(s[pos] == 'E') k = 0;
        else if(s[pos] == 'S') k = 1;
        else if(s[pos] == 'W') k = 2;
        else k = 3;
        for(int i = 1; i <= 3; i++) {
            x += dx[k];
            y += dy[k];
            v.push_back({x, y});
            sf(sf, s, pos+1, x, y, v, p);
        }
    };
};
```

```

vector<pair<int, int>> a1, a2;
vector<vector<pair<int, int>>> p1, p2;
a1.push_back({x1, y1});
a2.push_back({x2, y2});
dfs(dfs, s1, 0, x1, y1, a1, p1);
dfs(dfs, s2, 0, x2, y2, a2, p2);
int ans = 0;
for(auto t1 : p1) {
    for(auto t2 : p2) {
        set<pair<int, int>> s;
        // 用 set 找出兩者的交集數量
        for(auto tt1 : t1) {
            s.insert(tt1);
        }
        int cnt = 0;
        for(auto tt2 : t2) {
            if(s.count(tt2)) {
                cnt++;
                s.erase(tt2);
            }
        }
        ans = max(ans, cnt);
    }
}
cout << ans << endl;
}

```

spoiler AC code 2

```

#include <bits/stdc++.h>
using namespace std;

int main() {
    int x1, y1;
    int x2, y2;
    cin >> x1 >> y1;
    cin >> x2 >> y2;
    string s1, s2;
    cin >> s1;
    cin >> s2;

    int dx[4] = {1, 0, -1, 0};
    int dy[4] = {0, -1, 0, 1};
    auto dfs = [&](auto sf, string s, int pos, int x, int y, vector<pair<int,
int>> v, vector<vector<pair<int,int>>> &p) -> void {

        if(pos == s.length()) {
            p.push_back(v);
            return;

```

```

    }
    int k;
    if(s[pos] == 'E') k = 0;
    else if(s[pos] == 'S') k = 1;
    else if(s[pos] == 'W') k = 2;
    else k = 3;
    for(int i = 1; i <= 3; i++) {
        x += dx[k];
        y += dy[k];
        v.push_back({x, y});
        sf(sf, s, pos+1, x, y, v, p);
    }
};

vector<pair<int, int>> a1, a2;
vector<vector<pair<int, int>>> p1, p2;
a1.push_back({x1, y1});
a2.push_back({x2, y2});
dfs(dfs, s1, 0, x1, y1, a1, p1);
dfs(dfs, s2, 0, x2, y2, a2, p2);
int ans = 0;
for(auto t1 : p1) {

    for(auto t2 : p2) {
        int t[105][105]{}; // 改用 array 紀錄重複格子也可以
        for(auto tt1 : t1) {
            t[tt1.first+20][tt1.second+20]=1;
        }
        int cnt = 0;
        for(auto tt2 : t2) {
            if(t[tt2.first+20][tt2.second+20] == 1) {
                t[tt2.first+20][tt2.second+20] = 2;
                cnt++;
            }
        }
        ans = max(ans, cnt);
    }
}
cout << ans << endl;
}

```

spoiler AC code 3

```

#pragma GCC optimize("Ofast,unroll-loops")

#include <cstdio>
#include <cstring>
#include <cassert>

```



```

#include <algorithm>
#include <set>
#include <map>
#include <iomanip>
#include <limits>
#include <iostream>
#include <chrono>
#include <queue>
#include <random>
#include <functional>
#include <vector>
#include <bitset>
#include <ranges>

using namespace std;

#ifdef LOCAL
#include "debug"
#else
#define debug(...)
#endif

using i64 = int64_t;
using u32 = uint32_t;
using u64 = uint64_t;
using usize = size_t;
using f64 = double;
using i128 = __int128_t;
using u128 = __uint128_t;

template <class T>
constexpr T infty = 0;
template <>
constexpr int infty<int> = int(1e9);
template <>
constexpr int64_t infty<int64_t> = int64_t(1e18);
template <>
constexpr __int128_t infty<__int128_t> = __int128_t(1e36);
template <>
constexpr double infty<double> = 1e30;

#define rep(i, a, b) for (int i = (a); i <= (b); ++i)
#define per(i, a, b) for (int i = (a); i >= (b); --i)
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define fi first
#define se second

template<class T, class S>
inline bool chmin(T &a, const S &b) {
    return (a > b ? a = b, true : false);
}

```

```

template<class T, class S>
inline bool chmax(T &a, const S &b) {
    return (a < b ? a = b, true : false);
}
int xa, ya, xb, yb;
string S, T;
int state[200][200];
void Move(int who, int x, int y, string &S, int msk) {
    state[x][y] |= who;
    for (const char &c : S) {
        const int step = msk % 3 + 1;
        msk /= 3;
        if (c == 'E') {
            rep (_, 1, step) state[++x][y] |= who;
        } else if (c == 'W') {
            rep (_, 1, step) state[--x][y] |= who;
        } else if (c == 'S') {
            rep (_, 1, step) state[x][--y] |= who;
        } else if (c == 'N') {
            rep (_, 1, step) state[x][++y] |= who;
        } else {
            assert(false);
        }
    }
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cin >> xa >> ya >> xb >> yb;
    cin >> S >> T;
    const int N = int(size(S));
    const int M = int(size(T));
    xa += 50;
    ya += 50;
    xb += 50;
    yb += 50;
    int msks = 1;
    rep (i, 1, N) msks *= 3;
    int mskt = 1;
    rep (i, 1, M) mskt *= 3;
    int ANS = 0;
    rep (i, 0, msks - 1) {
        rep (j, 0, mskt - 1) {
            memset(state, 0, sizeof state);
            Move(1, xa, ya, S, i);
            Move(2, xb, yb, T, j);
            int same = 0;
            rep (x, 0, 100) rep (y, 0, 100) if (state[x][y] == 3) same++;
            chmax(ANS, same);
        }
    }
    cout << ANS << '\n';
}

```

```
return 0;  
}
```