

综合练习

人员

秦显森、刘闯速、赵熙羽、杨瑾硕、于子珈 到课, 高健桓、孙靖轲 线上

上周作业检查

https://cppoj.kids123code.com/contest/106

2025-0622 周日10:30 (综合练习)									
刷新									
#	用户名	姓名	总分	选择分	编程分	时间	A	B	C
1	chengzihao	程梓豪	200	0	200	340	100	100	
2	gaojianhuan	高健桓	200	0	200	418	100	100	
3	yuzijia	于子珈	200	0	200	420	100	100	0
4	zhaoxiyu	赵熙羽	200	0	200	421	100	100	0
5	qinxiansen	秦显森	200	0	200	438	100	100	0
6	sunjingke	孙靖轲	200	0	200	478	100	100	0
7	niutongze	牛同泽	200	0	200	523	100	100	
8	suiyuhan	隋钰涵	100	0	100	65	100		
9	liuchuangsu	刘闯速	100	0	100	80	100		
10	yangjinshuo	杨瑾硕	100	0	100			100	

作业

https://cppoj.kids123code.com/contest/133 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 B C 两道题稍微难一些, 同学们课上基本都没写完 C, 只写了一部分, 课下要抽时间把 C 题补完。

课堂内容

最后的迷宫

把终点向外延伸的 8 个方向能访问到的地方都打上标记, 然后从起点跑 bfs, 看最快什么时候能跑到标记点

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

using namespace std;

const int maxn = 16384 + 5;
vector<char> vec[maxn];
vector<int> dis[maxn];
vector<bool> st[maxn];
int n, m;
```

```

struct node {
    int x, y;
};
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};
int dx2[] = {-1, -1, -1, 0, 0, 1, 1, 1};
int dy2[] = {-1, 0, 1, -1, 1, -1, 0, 1};

int bfs(int sx, int sy, int ex, int ey) {
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) dis[i][j] = -1, st[i][j] = false;
    }

    for (int i = 0; i < 8; ++i) {
        int x = ex, y = ey;
        while (x>=1 && x<=n && y>=1 && y<=m && vec[x][y]=='0') {
            st[x][y] = true;
            x += dx2[i], y += dy2[i];
        }
    }

    queue<node> q; q.push({sx, sy}); dis[sx][sy] = 0;
    while (!q.empty()) {
        node u = q.front(); q.pop();
        int x = u.x, y = u.y;
        if (st[x][y]) return dis[x][y];
        for (int i = 0; i < 4; ++i) {
            int nx = x+dx[i], ny = y+dy[i];
            if (nx>=1 && nx<=n && ny>=1 && ny<=m && vec[nx][ny]=='0' && dis[nx][ny]==-1)
        {
            q.push({nx, ny}); dis[nx][ny] = dis[x][y] + 1;
        }
    }
}

return -1;
}

int main()
{
    cin >> n >> m;
    for (int i = 0; i <= n+2; ++i) {
        for (int j = 0; j <= m+2; ++j) {
            vec[i].push_back(' ');
            dis[i].push_back(0);
            st[i].push_back(false);
        }
    }

    for (int i = 1; i <= n; ++i) {
        string str; cin >> str; str = " " + str;
        for (int j = 1; j <= m; ++j) vec[i][j] = str[j];
    }

    while (true) {

```

```

        int sx, sy, ex, ey; cin >> ex >> ey >> sx >> sy;
        if (!sx) break;
        int t = bfs(sx, sy, ex, ey);
        if (t == -1) cout << "Poor Harry" << endl;
        else cout << t << endl;
    }
    return 0;
}

```

移动

简单题, set 套 pair 维护到过哪些点即可

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    set<pair<int,int>>st;

    int n; cin >> n;
    st.insert({0,0});
    string str; cin >> str;
    int x = 0, y = 0;
    for (char i : str) {
        if (i == 'U') ++y;
        else if (i == 'D') --y;
        else if (i == 'L') --x;
        else ++x;

        if (st.count({x,y})) { cout << "Yes" << endl; return 0; }
        st.insert({x,y});
    }
    cout << "No" << endl;
    return 0;
}

```

B3833 [NICA #2] 爱与不爱

把每一项变成对应的 2 的幂次方, 然后用一个 multiset 维护, 每次取最大值和最小值, 让最大值除2, 最小值乘2, 看能否让总和变小, 直到不能让总和变小为止

```

#include<bits/stdc++.h>
using namespace std;
const int maxn=1e5+20;
multiset<int>mt;
int change(int x)
{

```

```

int ans=0;
while(x/2>=1)
{
    ans++;
    x/=2;
}
int sum=1;
for(int i=1;i<=ans;++i){
    sum*=2;
}
return sum;
}
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;++i){
        int q;cin>>q;
        mt.insert(change(q));
    }
    while(1){
        int tmin=*mt.begin();
        int tmax=*mt.rbegin();
        if(tmin<tmax/2)
        {
            mt.erase(mt.find(tmin));
            mt.erase(mt.find(tmax));
            mt.insert(tmin*2);
            mt.insert(tmax/2);
        }
        else break;
    }
    long long cnt=0;
    for(auto i:mt){
        cnt+=i;
    }
    cout<<cnt;
    return 0;
}

```

[USACO16OPEN] Diamond Collector S

维护 ml 和 mr 两个数组, ml[i] 代表以 i 结尾时, 往左找最远能找到哪个位置; mr[i] 代表以 i 开头时, 往右找最远能找到哪个位置, 这个过程可以用 二分查找 来做

然后枚举一个分界点 i, 此时可以以 i 作为分界线, 在左边找一段长的, 在右边找一段长度

此时, 就是看以 1~i 结尾中, 哪一段最长; 还有以 i+1~n 开头, 哪一段最长

这里可以用一个 前缀最大值 和 后缀最大值 数组维护, 就可以 O(1) 求了

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

using namespace std;

const int maxn = 50000 + 5;
int w[maxn], mr[maxn], ml[maxn];
int pmax[maxn], smax[maxn];

int main()
{
    int n, k; cin >> n >> k;
    for (int i = 1; i <= n; i++) cin >> w[i];
    sort(w+1, w+n+1);

    for (int i = 1; i <= n; ++i) {
        int r = upper_bound(w+1, w+n+1, w[i]+k) - w - 1;
        int l = lower_bound(w+1, w+n+1, w[i]-k) - w;
        r = min(r, n), l = max(l, 1);
        mr[i] = r, ml[i] = l;
    }

    for (int i = 1; i <= n; ++i) pmax[i] = max(pmax[i], i - ml[i] + 1);
    for (int i = n; i >= 1; --i) smax[i] = max(smax[i+1], mr[i] - i + 1);

    int res = 0;
    for (int i = 1; i <= n-1; ++i) res = max(res, pmax[i] + smax[i+1]);
    cout << res << endl;
    return 0;
}
```