

综合练习

人员

田心一、纪博涵、蒋叔璋、初锦阳、刘宸熙、赵书梵 到课, 高健桓、柳力玮 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2172>

#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	zhaoshufan	赵书梵	600	1246	100	100	100	100	100	100
2	yuanzhao	苑钊	600	1581	100	100	100	100	100	100
3	gaojianhuan	高健桓	600	1792	100	100	100	100	100	100
4	tianxinyi	田心一	600	2129	100	100	100	100	100	100
5	liruihan	李瑞涵	600	2507	100	100	100	100	100	100
6	liuchenxi	刘宸熙	500	1215	100	100	100	100	100	100
7	chujinnyang	初锦阳	400	973	100	100	100	100		
8	liuliwei	柳力玮	400	1113	100	100	100	100		
9	jiangshuzhang	蒋叔璋	400	1264	100	100	100	100		
	jibohan	纪博涵	400	1471	100	100	100		100	

您正在共享屏幕 结束共享

本周作业

<https://cppoj.kids123code.com/contest/2241> (课上讲了 A ~ D 题, 课后作业是 E 题)

课堂表现

今天的题相对复杂一些, 尤其是 D 题, 需要同学们认真画画图做一做。

课堂内容

Mzc和男家丁的游戏 (上周作业)

从字符 'm' 开始做 bfs, 求到字符 'd' 的最短路

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

using namespace std;

const int maxn = 2000 + 5;
char s[maxn][maxn];
int w[maxn][maxn];
struct node {
    int x, y;
```

```

};

int dx[] = {-1, 1, 0, 0};
int dy[] = {0, 0, -1, 1};

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

    memset(w, -1, sizeof(w));
    queue<node> q;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            if (s[i][j] == 'm') q.push({i, j}), w[i][j] = 0;
        }
    }

    while (!q.empty()) {
        node u = q.front(); q.pop();
        int x = u.x, y = u.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 && s[nx][ny]!='#' && w[nx][ny]==-1) {
                q.push({nx, ny}); w[nx][ny] = w[x][y] + 1;
            }
        }
    }

    int res = 1e9;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            if (s[i][j] == 'd' && w[i][j] != -1) res = min(res, w[i][j]);
        }
    }
    if (res == 1e9) cout << "No Way!" << endl;
    else cout << res << endl;
    return 0;
}

```

奇怪的电梯

针对每个点 i , 可以花一步到达点 $i+w[i]$ 和点 $i-w[i]$, 从点 A 开始做 bfs, 求到点 B 的最短路即可

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 200 + 5;
int w[maxn], dis[maxn];

int main()

```

```

{
    int n, A, B; cin >> n >> A >> B;
    for (int i = 1; i <= n; ++i) cin >> w[i];

    memset(dis, -1, sizeof(dis));
    queue<int> q; q.push(A); dis[A] = 0;
    while (!q.empty()) {
        int u = q.front(); q.pop();
        if (u+w[u]<=n && dis[u+w[u]]== -1) dis[u+w[u]]=dis[u]+1, q.push(u+w[u]);
        if (u-w[u]>=1 && dis[u-w[u]]== -1) dis[u-w[u]]=dis[u]+1, q.push(u-w[u]);
    }
    cout << dis[B] << endl;
    return 0;
}

```

[USACO1.5] 八皇后 Checker Challenge

dfs + 剪枝

从第 1 行往第 n 行进行 dfs 搜索, 在第 u 行时, 枚举第 1 列到第 n 列, 看每个格子是否能放

如果能放, 记录下这个格子的信息, 并往后继续进行 dfs

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 15;
int col[maxn], l1[2*maxn], l2[2*maxn];
int path[maxn];
int res = 0;

void dfs(int u, int n) {
    if (u == n+1) {
        if (res < 3) {
            for (int i = 1; i <= n; i++) cout << path[i] << " ";
            cout << endl;
        }
        ++res;
        return;
    }

    for (int i = 1; i <= n; i++) {
        if (col[i] || l1[u+i] || l2[i-u+n]) continue;

        path[u] = i;
        col[i] = l1[u+i] = l2[i-u+n] = true;
        dfs(u+1, n);
        col[i] = l1[u+i] = l2[i-u+n] = false;
    }
}

```

```

int main()
{
    int n;
    cin >> n;
    dfs(1, n);
    cout << res << endl;
    return 0;
}

```

[IOI 1994 / USACO1.5] 数字三角形 Number Triangles

记忆化搜索, $\text{dfs}(x,y)$, 用来计算从 (x,y) 这个点往下搜时, 能获得的最大路径和是多少

把搜到的结果记在 $f[x][y]$ 中, 时间复杂度为 n^2 级别

```

#include <iostream>
#include <cstring>

using namespace std;

const int maxn = 1000 + 5;
int w[maxn][maxn], f[maxn][maxn];
int n;

int dfs(int x, int y) {
    if (x == n+1) return 0;
    if (f[x][y] != -1) return f[x][y];
    f[x][y] = max(dfs(x+1,y),dfs(x+1,y+1)) + w[x][y];
    return f[x][y];
}

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

    memset(f, -1, sizeof(f));
    cout << dfs(1, 1) << endl;
    return 0;
}

```

[USACO3.4] 美国血统 American Heritage

中序 + 前序求后序

$\text{dfs}(l1, r1, l2, r2)$: 针对 a 数组的 $l1 \sim r1$ 部分和 b 数组的 $l2 \sim r2$ 部分, 求对应的后序是什么

b[l2] 就是当前的根, 在 a 数组中找到 b[l2] 这个字符, 根据这个根, 可以把整棵树分成两部分, 继续进行 dfs 搜索即可

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

using namespace std;

const int maxn = 26 + 5;
char a[maxn], b[maxn];

int w_find(int l, int r, char x) {
    for (int i = l; i <= r; ++i) {
        if (a[i] == x) return i;
    }
    return 0;
}

void dfs(int l1, int r1, int l2, int r2) {
    if (l1 > r1) return;
    if (l1 == r1) { cout << a[l1]; return; }

    char x = b[l2];
    int pos = w_find(l1, r1, x);
    int l_len = pos - l1;
    dfs(l1, pos-1, l2+1, l2+l_len);
    dfs(pos+1, r1, l2+l_len+1, r2);
    cout << x;
}

int main()
{
    cin >> (a+1) >> (b+1);
    dfs(1, strlen(a+1), 1, strlen(b+1));
    cout << endl;
    return 0;
}
```