

dp练习

人员

于家瑞、陶汇笙、洪晨栋、洪晨棋、于霄龙、郭栩睿、崔宸赫 到课, 王恩泽 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1476>

王向东老师周六十点半C+++dp入门												
#	用户名	姓名	编程分	时间	A	B	C	D	E	F	G	
1	taohuisheng	陶汇笙	700	666	100	100	100	100	100	100	100	
2	yujarui	于家瑞	700	1005	100	100	100	100	100	100	100	
3	hongchendong	洪晨栋	550	551	100	100	100	100	100	50		
4	hongchenqi	洪晨棋	500	541	100	100	100	100	100			
5	yuxiaolong	于霄龙	500	542	100	100	100	100	100			
6	guoxurui	郭栩睿	500	555	100	100	100	100	100			
7	cuichenhe	崔宸赫	400	390	100	100	100	100				
8	wangenze	王恩泽	300	168			100	100	100			

本周作业

<https://cppoj.kids123code.com/contest/1583> (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

今天课上这几道题, 好几位同学在听懂之后都不能第一时间做对, 说明同学们写代码时细节把握不好, 还是不够细心, 要养成边做题边画图仔细分析的习惯。

课堂内容

[NOIP 2002 普及组] 过河卒 (上周作业)

遇到被马覆盖的点不走, 否则 方案数 可以从 正上方 或者 左方 转移而来

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

using namespace std;

typedef long long LL;
const int maxn = 20 + 5;
LL w[maxn][maxn];

bool check(int x, int y, int i, int j) {
    if (x==i && y==j) return true;
    if (x-2==i && y-1==j) return true;
    if (x-2==i && y+1==j) return true;
    if (x+2==i && y-1==j) return true;
    if (x+2==i && y+1==j) return true;
    if (x-1==i && y-2==j) return true;
    if (x+1==i && y-2==j) return true;
    if (x-1==i && y+2==j) return true;
    if (x+1==i && y+2==j) return true;
    return false;
}
```

```

int dx = abs(x - i), dy = abs(y - j);
if (dx > dy) swap(dx, dy);
return dx==1 && dy==2;
}

int main()
{
    int n, m, x, y; cin >> n >> m >> x >> y;
    for (int i = 0; i <= n; ++i) {
        for (int j = 0; j <= m; ++j) {
            if (!i && !j) w[i][j] = 1;
            else {
                if (check(x, y, i, j)) continue;
                if (i >= 1) w[i][j] += w[i-1][j];
                if (j >= 1) w[i][j] += w[i][j-1];
            }
        }
    }
    cout << w[n][m] << endl;
    return 0;
}

```

红牌

从左往右走, 所以应该先枚举列, 先枚举行

$f[i][j]$ 代表走到第 i 行第 j 列时, 最少花多少天

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2000 + 5;
int w[maxn][maxn], f[maxn][maxn];

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

    memset(f, 0x3f, sizeof(f));
    for (int i = 1; i <= n; ++i) f[i][1] = w[i][1];

    for (int j = 2; j <= m; ++j) {
        for (int i = 1; i <= n; ++i) {
            if (i == 1) f[i][j] = min(f[i][j-1], f[n][j-1]) + w[i][j];
            else f[i][j] = min(f[i-1][j-1], f[i][j-1]) + w[i][j];
        }
    }
}

```

```

int res = 1e9;
for (int i = 1; i <= n; ++i) res = min(res, f[i][m]);
cout << res << endl;
return 0;
}

```

封印

$f[i]$: 1->i 最小花费

答案: $f[n]$

从左往右求 $f[i]$

```

for (int i = 1; i <= n; i++) {
    求  $f[i]$  ?
    枚举 i 前面的 j: 先从1到j-1花费最小, 再从j到i
}

```

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 10000 + 5;
int w[maxn], p[maxn];
LL f[maxn];

int get_sum(int l, int r) { return (l<=r ? p[r]-p[l-1] : 0); }

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

    memset(f, 0x3f, sizeof(f));
    f[0] = 0;
    for (int i = 1; i <= n; ++i) {
        f[i] = f[i-1] + (LL)w[i]*n*n;
        for (int j = 1; j < i; j++) {
            if (w[j]+w[i] <= t) f[i] = min(f[i], f[j-1] + ((LL)w[j]+w[i])*get_sum(j,i));
        }
    }
    cout << f[n] << endl;
    return 0;
}

```

两个排列的最长公共子序列

这道题目前得到 50 分即可, 满分做法以后会讲

$f[i][j]$: 把第一个数组的前 i 项, 和第二个数组的前 j 项变得一样时, 最少需要几次操作

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

using namespace std;

const int maxn = 1e3 + 5;
int a[maxn], b[maxn];
int f[maxn][maxn];

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

    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= n; ++j) {
            if (a[i] == b[j]) f[i][j] = f[i-1][j-1] + 1;
            else f[i][j] = max(f[i-1][j], f[i][j-1]);
        }
    }

    cout << f[n][n] << endl;
    return 0;
}
```