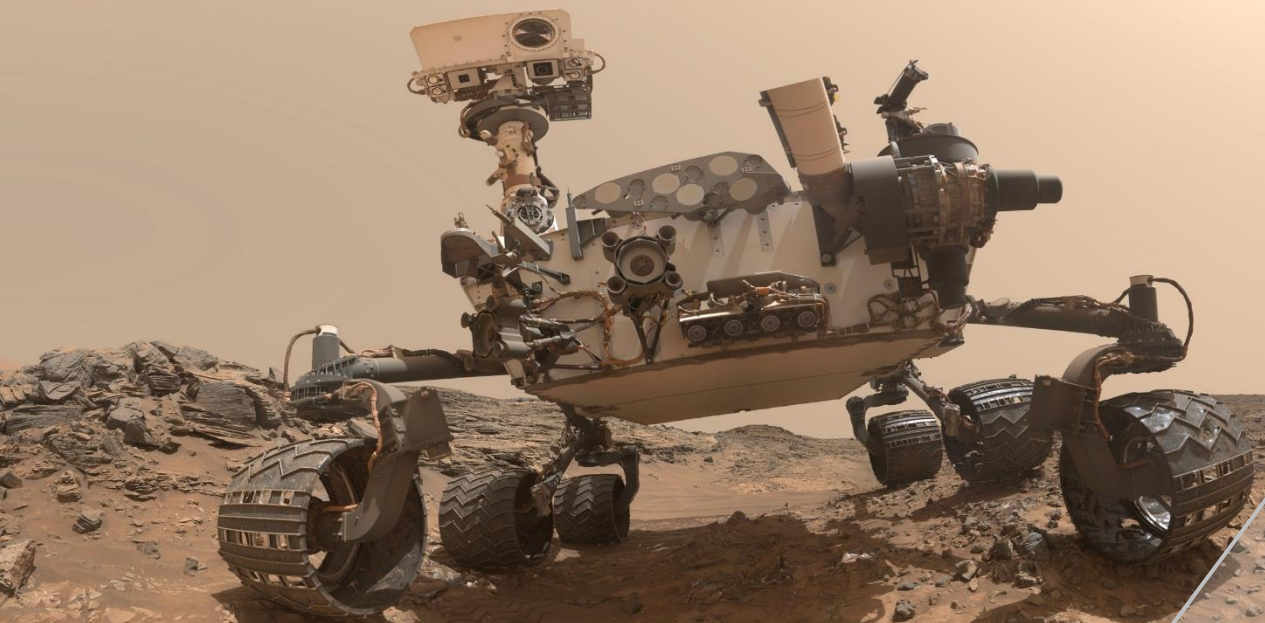
CS101



信奥算法

课件下载地址：
http://pan.baidu.com/s/1o885tz0

作业网站：
http://120.132.18.213:8080/thrall-web/main#home

二维递推问题

综合练习

棋盘路径计数 - 无障碍

在棋盘格上，小明站在第1行第1列的位置（左上角），罗马在第n行第m列的位置（右下角）。小明每一步只可以向右走一格或者向下走一格。输入n和m (1<=m,n<=20)，输出小明有多少种方法可以走到罗马。

输入样例：
2 2

输出样例：
2

输入样例：
4 3

输出样例：
10

棋盘路径计数 - 无障碍

f[i][j]表示 到第i行第j列有多少种走法

棋盘路径计数 - 无障碍

f[i][j]表示 到第i行第j列有多少种走法

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	1	1	1
i=2	0	1	2	3	4
i=3	0	1	3	6	10
i=4	0	1	4	10	20

缓冲带

缓冲带

棋盘路径计数 - 无障碍

f[i][j]表示 到第i行第j列有多少种走法

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	1	1	1
i=2	0	1	2	3	4
i=3	0	1	3	6	10
i=4	0	1	4	10	20

缓冲带

缓冲带

$f[1][1] = 1$

初始条件

(i,j)不是(1,1)时 $f[i][j] = f[i-1][j] + f[i][j-1]$

递推方程

```cpp
#include<iostream>
#define M 21
using namespace std;
long long f[M][M],n,m,i,j;
int main(){
    cin>>n>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++)
            if(i==1&&j==1) f[1][1]=1;
            else f[i][j]=f[i-1][j]+f[i][j-1];
    cout<<f[n][m]<<endl;
    return 0;
}
```

棋盘路径计数 - 有障碍

在棋盘格上，小明站在第1行第1列的位置（左上角），目的地在第5行第5列的位置（右下角）。小明每一步只可以向右走一格或者向下走一格。
输入5行5列的棋盘，0表示无障碍可以通行，#表示此处有障碍，输出小明有多少种方法可以走到目的地。

输入样例：
00000
0##0#
0##00
0###0
00000

输出样例：
2

输入样例：
00000
##0##
##0##
##000
##000

输出样例：
3

棋盘路径计数 - 有障碍

在棋盘格上，小明站在第1行第1列的位置（左上角），目的地在第5行第5列的位置（右下角）。小明每一步只可以向右走一格或者向下走一格。
输入5行5列的棋盘，0表示无障碍可以通行，#表示此处有障碍，输出小明有多少种方法可以走到目的地。

输入样例：
00000
0##0#
0##00
0###0
00000

输入样例：
00000
##0##
##0##
##000
##000

char d[6][6];
数组d保存整张地图

d[i][j]表示 第i行第j列的字符

输出样例：
2

输出样例：
3

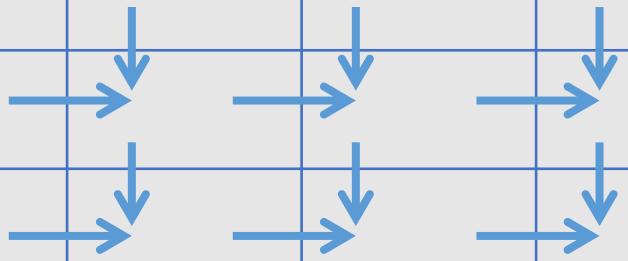
棋盘路径计数 - 有障碍

f[i][j]表示 到第i行第j列有多少种走法

棋盘路径计数 - 有障碍

f[i][j]表示 到第i行第j列有多少种走法

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0				
i=2	0				
i=3	0				



缓冲带

缓冲带

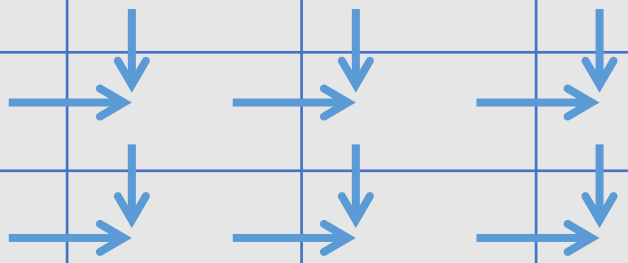
$$f[1][1] = (d[1][1] == \text{'0'})$$

初始条件

棋盘路径计数 - 有障碍

f[i][j]表示 到第i行第j列有多少种走法

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0				
i=2	0				
i=3	0				



缓冲带

缓冲带

$$f[1][1] = (d[1][1] == \text{'0'})$$

初始条件

(i,j)不是(1,1)	若$d[i][j] = \text{'\#'}$	$f[i][j] = 0$
	若$d[i][j] = \text{'0'}$	$f[i][j] = f[i-1][j] + f[i][j-1]$

递推方程

```cpp
#include<iostream>
#define N 5
using namespace std;
int f[6][6],i,j;
char d[6][6];
int main(){
    for(i=1;i<=N;i++)
        for(j=1;j<=N;j++)
            cin>>d[i][j];
    for(i=1;i<=N;i++)
        for(j=1;j<=N;j++)
            if(i==1&&j==1) f[1][1]=(d[i][j]=='0');
            else if(d[i][j]=='#') f[i][j]=0;
            else f[i][j]=f[i-1][j]+f[i][j-1];
    cout<<f[N][N]<<endl;
    return 0;
}
```

棋盘最优路径 - 无障碍

在棋盘格子里有些金币。小明站在第1行第1列（左上角），目的地在第5行第5列（右下角），可以取走路过每一格里的金币。小明每一步只可以向右走一格或者向下走一格。
输入5行5列的棋盘，每格数字表示金币个数。输出小明走到目的地最多能拿多少金币。

输入样例：
00000
01104
01100
01110
00000

输出样例：
6

输入样例：
10000
00000
00202
03000
00001

输出样例：
6

棋盘最优路径 - 无障碍

在棋盘格子里有些金币。小明站在第1行第1列（左上角），目的地在第5行第5列（右下角），可以取走路过每一格里的金币。小明每一步只可以向右走一格或者向下走一格。
输入5行5列的棋盘，每格数字表示金币个数。输出小明走到目的地最多能拿多少金币。

输入样例：
00000
01104
01100
01110
00000

输出样例：
6

输入样例：
10000
00000
00202
03000
00000

输出样例：
5

int d[6][6];
数组d保存整张地图

d[i][j]表示 第i行第j列金币数

棋盘最优路径 - 无障碍

g[i][j]表示 到第i行第j列时最多拿走多少金币

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0				
i=2	0				
i=3	0				

缓冲带

缓冲带

$$g[1][1] = d[1][1]$$

初始条件

(i,j)不是(1,1)	$g[i][j] = max(g[i-1][j], g[i][j-1]) + d[i][j]$

递推方程

```cpp
#include<iostream>
#define N 5
using namespace std;
int d[6][6],g[6][6],i,j;
int main(){
    for(i=1;i<=N;i++)
        for(j=1;j<=N;j++){
            char ch; cin>>ch;
            d[i][j]=ch-'0';
        }
    for(i=1;i<=N;i++)
        for(j=1;j<=N;j++)
            if(i==1&&j==1) g[1][1]=d[i][j];
            else g[i][j]=max(g[i-1][j],g[i][j-1])+d[i][j];
    cout<<g[N][N]<<endl;
    return 0;
}
```