

CS101

A Mars rover, likely a Curiosity rover, is shown on a rocky, reddish-brown landscape. The rover is white with various instruments and cameras. It has six large, treaded wheels. The background shows a hazy, orange sky and distant hills. The overall scene is a typical Mars surface environment.

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>

作业1：集体照

```
1  #include<iostream>
2  using namespace std;
3  long long i,n,f[31];
4  int main(){
5      cin>>n;
6      f[0]=1;
7      for(i=1;i<=n;i++)
8          f[i]=f[i-1]*(4*i-2)/(i+1);
9      cout<<f[n]<<endl;
10     return 0;
11 }
```

作业2：条条大路通罗马

```
1 #include<iostream>
2 #define M 21
3 using namespace std;
4 long long f[M][M],n,m,i,j;
5 int main(){
6     cin>>n>>m;
7     for(j=1;j<=m;j++) f[1][j]=1;
8     for(i=2;i<=n;i++){
9         f[i][1]=1;
10        for(j=2;j<=m;j++)
11            f[i][j]=f[i-1][j]+f[i][j-1];
12    }
13    cout<<f[n][m]<<endl;
14    return 0;
15 }
```

作业2： 条条大路通罗马x

```
1  #include<iostream>
2  #define M 21
3  using namespace std;
4  long long f[M][M],n,m,i,j;
5  int main(){
6      cin>>n>>m;
7      f[1][1]=1;
8      for(i=1;i<=n;i++)
9          for(j=1;j<=m;j++)
10             if(i==1&&j==1) continue;
11             else f[i][j]=f[i-1][j]+f[i][j-1];
12      cout<<f[n][m]<<endl;
13      return 0;
14 }
```

作业3：野兽出没

```
1  #include<iostream>
2  #define M 21
3  using namespace std;
4  long long f[M][M],n,m,x,y,i,j;
5  int main(){
6      cin>>n>>m>>x>>y;
7      f[1][1]=1;
8      for(j=1;j<=m;j++) //第1行
9          if(x==1&&y==j) break;
10         else f[1][j]=1;
11     for(i=2;i<=n;i++) //第2到n行
12         for(j=1;j<=m;j++){
13             if(x==i&&y==j) continue;
14             if(j==1) f[i][1]=1;
15             else f[i][j]=f[i-1][j]+f[i][j-1];
16         }
17     cout<<f[n][m]<<endl;
18     return 0;
19 }
```

作业3: 野兽出没x

```
1  #include<iostream>
2  #define M 21
3  using namespace std;
4  long long f[M][M],n,m,x,y,i,j;
5  int main(){
6      cin>>n>>m>>x>>y;
7      f[1][1]=1; f[x][y]=0;
8      for(i=1;i<=n;i++)
9          for(j=1;j<=m;j++){
10             if(i==1&&j==1) continue;
11             if(x==i&&y==j) continue;
12             f[i][j]=f[i-1][j]+f[i][j-1];
13         }
14     cout<<f[n][m]<<endl;
15     return 0;
16 }
```

递推问题

综合练习

递推算法 规律小结

当i为0时	$f[0] = 1$
当i为1时	$f[1] = 1$

初始
条件

i为0,1,2
等等情况

当i大于1时	$f[i] = f[i - 1] + f[i - 2]$
--------	------------------------------

递推
方程

f[i]可以由已经求解出的
f[i-1],f[i-2]等等推得

例题：走楼梯

小明共有 n 级楼梯要走，每一步他只能向上走2级楼梯或者1级楼梯，请问共有多少种不同的走法。例如 $n=3$ 时共有三种走法：可以先走1级再走2级，或者先走2级再走1级，或者每次都走1级。输入一个正整数 n （ $n \leq 50$ ），输出一个正整数代表共有多少种走法可以正好走完 n 级台阶。

输入样例：

4

输入样例：

3

输出样例：

5

输出样例：

3

一步一步
从低到高走

走楼梯：解法1

$f[i]$ 代表走到第*i*级共有多少种方法

当 <i>i</i> 为1时	$f(1) = 1$
当 <i>i</i> 为2时	$f(2) = 2$

初始
条件

当 <i>i</i> 大于2时	$f(i) = f(i - 1) + f(i - 2)$
-----------------	------------------------------

递推
方程

来到第*i*级的方式有两种：
可以走1级来
也可以走2级来

走楼梯：解法2

$f[i]$ 代表走到第*i*级共有多少种方法

初始
条件

当*i*为0时

$$f[0] = 1$$

递推
方程

当*i*大于0时

$$f[i] = f[i - 1] \mid i \geq 1 \\ + f[i - 2] \mid i \geq 2$$

若*i* ≥ 1，可走1级来

若*i* ≥ 2，可走2级来

递推问题分类

计数问题：统计所有可能性的个数

优化问题：在所有可能性中哪个最好

例题：打游戏

小明沉溺于打手机游戏无法自拔，严重影响了他的身体健康。妈妈虽然仍然允许他适度地打游戏，但是加了一条限制条件：不能连续两天都打游戏。

输入第一行是一个正整数 n 代表天数，第二行为 n 个正整数代表这连续 n 天每天可以打游戏的小时数。输出一个正整数，代表小明这些天最多能打多少小时时间的游戏。 $n \leq 100$

输入样例：

4

3 1 1 2

输出样例：

5

一天一天
按时间顺序打

打游戏：解法1

$f[i]$ 代表第*i*天打游戏时前*i*天最多打多久

第*i*天可以打 $x[i]$ 小时

当 <i>i</i> 为0时	$f[0] = 0$
当 <i>i</i> 为1时	$f[1] = x[1]$
当 <i>i</i> 为2时	$f[2] = x[2]$

初始
条件

打游戏：解法1

$f[i]$ 代表第*i*天打游戏时前*i*天最多打多久
第*i*天可以打 $x[i]$ 小时

当 <i>i</i> 为0时	$f[0] = 0$	初始 条件
当 <i>i</i> 为1时	$f[1] = x[1]$	
当 <i>i</i> 为2时	$f[2] = x[2]$	

当 <i>i</i> 大于3时	$f[i] = \max(f[i - 2], f[i - 3]) + x[i]$	最优性 递推方程
-----------------	--	-------------

因为第*i*天要打游戏，所以前一次打有两种：
第*i*-2天，或第*i*-3天

$$ans = \max(f[n], f[n - 1])$$

最优
答案

打游戏：解法2

$g[i]$ 代表 前*i*天最多打多久

第*i*天可以打 $x[i]$ 小时

当 <i>i</i> 为0时	$g[0] = 0$	初始 条件
当 <i>i</i> 为1时	$g[1] = x[1]$	

当 <i>i</i> 大于2时	$g[i] = \max(g[i - 1], g[i - 2] + x[i])$	最优性 递推方程
-----------------	--	-------------

有两种可能性：
第*i*天不打，或第*i*天打

$ans = g[n]$

最优
答案

例题：满地铜币

地上有 n 个格子，每个里面都有一些铜币。小明目前站在第一格，他每走一步能前进2格，或者3格。请问他最多能捡到几个铜币。输入第一行为正整数 n ，第二行为 n 个正整数代表每格铜币个数。输出小明最多能捡多少个铜币。 $n \leq 1000$

输入样例：

6

1 1 2 3 4 5

输出样例：

9

一格一格
按顺序捡铜币

满地铜币

$f[i]$ 代表走到第*i*格时最多已经捡几个铜币
第*i*格有 $x[i]$ 个铜币

当*i*为1时

$$f[1] = x[1]$$

初始
条件

当*i*大
于1时

$$f[i] = \max(f[i-2] \mid i > 2, f[i-3] \mid i > 3) + x[i]$$

最优性
递推方程

有两种可能性：
如果*i*>2可以从第*i*-2格来，
如果*i*>3可以从第*i*-3格来

$$ans = \max(f[n], f[n-1], f[n-2], \dots, f[1])$$

最优
答案