

# 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/1nu6kYkL>

# 作业01

```
7  cin>>n>>m;
8  for(j=1;j<=m;j++) cin>>w[j];
9  f[0]=0;
10 for(i=1;i<=n;i++){
11     for(j=1;j<=m;j++){
12         if(i<w[j]) x[j]=0;
13         else x[j]=f[i-w[j]]+w[j];
14     }
15     f[i]=x[1];
16     for(j=2;j<=m;j++) f[i]=max(f[i],x[j]);
17 }
18 cout<<f[n]<<endl;
```

# 作业02

```
7  cin>>C>>n;
8  for(int i=1;i<=n;++i) cin>>w[i]>>v[i];
9  // 计算f[i][j]从前i件物品中选出总重量不超过j的物品的最大价值
10 for (int i=1; i<=n; ++i)    //循环查看物品i
11     for (int j=0; j<=C; ++j) { //循环查看背包剩余重量j
12         if (j<w[i])    // 物品i太重, 无法放入
13             f[i][j]=f[i-1][j];
14         else            // 比较两种决策: 物品i可以放, 或者不放
15             f[i][j]=max(f[i-1][j], f[i-1][j - w[i]] + v[i]);
16     }
17 cout<<f[n][C]<<endl;
```

# 作业03

```
1  #include<iostream>
2  #define MAXT 1005
3  #define MAXM 105
4  using namespace std;
5  int t[MAXM],v[MAXM],f[MAXM][MAXT],i,j,T,M;
6  int main(){
7      cin>>T>>M;
8      for(i=1;i<=M;i++) cin>>t[i]>>v[i];
9      for(i=1;i<=M;i++)
10         for(j=1;j<=T;j++) {
11             f[i][j]=f[i-1][j];
12             if(t[i]<=j)
13                 f[i][j]=max(f[i][j],f[i-1][j-t[i]]+v[i]);
14         }
15     cout<<f[M][T]<<endl;
16     return 0;
17 }
```

# 回顾：01背包问题的变种

给定总的容量/时间/预算限制

有若干种备选物品/草药，每种只能用0次或1次

每次装包/购买/使用，都会消耗/费用

最大化总价值/收益/满足度

# 01背包：最大价值

小偷来你家，他带的包只能装 $n$ 公斤的财物。你家有 $m$ 种财物，分别重 $w_1, w_2, \dots, w_m$ 公斤，价值分别为 $v_1, v_2, \dots, v_m$ 求带走**最大的总价值**是多少



3斤, ¥5000,



1斤, ¥3000,



2斤, ¥3500,



只能装4斤



# 01背包：最大重量

小偷去银行，他带的包只能装 $n$ 公斤的财物。有 $m$ 种金块，分别重 $w_1, w_2, \dots, w_m$ 公斤，求带走**最大的重量**是多少





# 凑数问题

思考：凑合问题和背包问题的相似点

# 多数凑和 - 总和尽量大

给定一个目标数 $a$ ，和 $n$ 个正整数 $x_1, x_2, \dots, x_n$ ，判断能否找到若干个数总和尽量大但不超过 $a$ 。每个数只能用一次。

输入样例：

10 3

3 4 8

输入样例：

5 2

8 9

输出样例：

8

输出样例：

0

# 多数凑和- 总和尽量大

给定一个目标数 $a$ ，和 $n$ 个正整数 $x_1, x_2, \dots, x_n$ ，判断能否找到若干个数总和尽量大但不超过 $a$ 。每个数只能用一次。

$f[i][j]$ 表示只用前 $i$ 个数中的数求和，  
总和不大于 $j$ 时最大是多少

当 $i$ 为0时 $f[0][j] = 0$

当 $j$ 为0时 $f[i][0] = 0$

$$f[i][j] = \max \{ f[i-1][j], f[i-1][j-x_j] + x_j \}$$

不用第 $i$ 个数

用第 $i$ 个数

# 多数凑和- 判断可行性

给定一个目标数 $a$ ，和 $n$ 个正整数 $x_1, x_2, \dots, x_n$ ，判断能否找到若干个数总和恰巧为 $a$ 。每个数只能用一次。

输入样例：

10 3

3 4 8

输入样例：

10 5

6 5 4 3 2

输出样例：

No

输出样例：

Yes

# 多数凑和- 判断可行性

给定一个目标数 $a$ ，和 $n$ 个正整数 $x_1, x_2, \dots, x_n$ ，判断能否找到若干个数总和恰巧为 $a$ 。每个数只能用一次。

$f[i][j]$ 表示只用前 $i$ 个数中的数求和，  
总和恰巧为 $j$ 是否可以  
可以用1表示，不可以用0表示

当 $i=0, j>0$ 时  $f[0][j] = 0$

当 $j=0$ 时  $f[i][0] = 1$

$$f[i][j] = f[i-1][j] \text{ or } f[i-1][j-x_j]$$

不用第 $i$ 个数

用第 $i$ 个数

# 思考题

如何利用“可行性问题”的解答，来回答“总和尽量大”的问题

# 多数凑和- 计数

给定一个目标数 $a$ ，和 $n$ 个正整数，请统计其中若干个数的总和恰巧为 $a$ 有几种可能的组合。每个数只能用一次。

输入样例：

10 3  
3 4 8

输入样例：

10 5  
6 5 4 3 2

输出样例：

0

输出样例：

2



# 多数凑和- 计数

给定一个目标数 $a$ ，和 $n$ 个正整数，请统计其中若干个数总和恰巧为 $a$ 有几种可能的组合。每个数只能用一次。

$f[i][j]$ 表示只用前 $i$ 个数中的数求和，  
总和恰巧为 $j$ 有几种可能的组合

当 $i=0, j>0$ 时  $f[0][j] = 0$

当 $j=0$ 时  $f[i][0] = 1$

$$f[i][j] = f[i-1][j] + f[i-1][j-x_j]$$

不用第 $i$ 个数

用第 $i$ 个数

# 多数凑和- 总和最接近目标

给定一个目标数 $a$ ，和 $n$ 个正整数，判断能否找到若干个  
数总和尽量接近目标 $a$ 。每个数只能用一次。

输入样例：

10 3

3 4 8

输入样例：

5 2

8 9

输出样例：

11

输出样例：

8

思考题：如何利用“可行性问题”的解答，来回答“总和最接近目标”的问题

# 动态规划常见问题分类

可行性判定问题

计数问题：共几种可能性

最优性问题：求最大/最小

# 棋盘路径 - 可行性问题

在 $n*m$ 的棋盘格上，有一些障碍物用#表示，如果每一步只可以向右走或者向下走一格，请问能否从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）。

输入样例：

```
3 4  
0000  
00##  
00#0
```

输入样例：

```
3 4  
0000  
00#0  
00#0
```

输出样例：

No

输出样例：

Yes

# 棋盘路径 - 可行性问题

在 $n*m$ 的棋盘格上，有一些障碍物用#表示，如果每一步只可以向右走或者向下走一格，请问能否从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）。

$f[i][j]$ 表示第 $i$ 行第 $j$ 列能否走到  
0代表可以，1代表不可以

当 $i=0$ 时  $f[0][j] = 0$

当 $j=0$ 时  $f[i][0] = 0$

$f[1][1] = 1$

此处有障碍物#

$f[i][j] = 0$

此处无障碍物

$f[i][j] = f[i-1][j] \text{ or } f[i][j-1]$

# 棋盘路径 - 计数问题

在 $n*m$ 的棋盘格上，有一些障碍物用#表示，如果每一步只可以向右走或者向下走一格，请问从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）一共有几条路径。

输入样例：

```
3 4  
0000  
00##  
00#0
```

输入样例：

```
2 4  
0000  
0#00
```

输出样例：

0

输出样例：

2

# 棋盘路径 - 计数问题

在 $n*m$ 的棋盘格上，有一些障碍物用#表示，如果每一步只可以向右走或者向下走一格，请问从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）一共有几条路径。

$f[i][j]$ 表示走到第 $i$ 行第 $j$ 列有几条路径

当 $i=0$ 时  $f[0][j] = 0$

当 $j=0$ 时  $f[i][0] = 0$

$f[1][1] = 1$

此处有障碍物#

$f[i][j] = 0$

此处无障碍物

$f[i][j] = f[i-1][j] + f[i][j-1]$



# 棋盘路径 - 最优路径

在 $n*m$ 的棋盘格上，有一些金块，每个用\$表示，如果每一步只可以向右走或者向下走一格，请问从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）最多捡到多少个金块。

输入样例：

3 4

0000

00\$\$

00\$0

输入样例：

2 4

00\$0

0\$00

输出样例：

2

输出样例：

1

# 棋盘路径 - 最优路径

在 $n*m$ 的棋盘格上，有一些金块，每个用\$表示，如果每一步只可以向右走或者向下走一格，请问从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）最多捡到多少个金块。

$f[i][j]$ 表示走到第 $i$ 行第 $j$ 列最多捡多少金块

当 $i=0$ 时  $f[0][j] = 0$

当 $j=0$ 时  $f[i][0] = 0$

当起始位置有金块  $f[1][1] = 1$  否则  $f[1][1] = 0$

此处无金块

$$f[i][j] = \max\{f[i-1][j], f[i][j-1]\}$$

此处有金块

$$f[i][j] = \max\{f[i-1][j], f[i][j-1]\} + 1$$

# 思考题：最短路径问题

# 棋盘路径 - 最短路径I

在 $n*m$ 的棋盘格上，有一些障碍物用#表示，如果每一步只可以向右走或者向下走一格，请问从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）最短要走几步。如果走不通输出-1

输入样例：

```
3 5  
0#000  
0#0#0  
000#0
```

输出样例：

-1

输入样例：

```
2 4  
0000  
0#00
```

输出样例：

4

# 棋盘路径 - 最短路径II

在 $n*m$ 的棋盘格上，有一些障碍物用#表示，如果每一步可以向上/向下/向左/向右走移动一格，请问从第1行第1列的位置（左上角）到第 $n$ 行第 $m$ 列的位置（右下角）最短要走几步。

输入样例：

```
3 5  
0#000  
0#0#0  
000#0
```

输出样例：

```
10
```

输入样例：

```
2 4  
0000  
0#00
```

输出样例：

```
4
```