

CS102

A Mars rover, likely a Curiosity rover, is shown on a rocky, orange-hued landscape. The rover is positioned in the center-left of the frame, facing right. It has six large, treaded wheels and a complex body with various instruments and cameras. The background shows a hazy, orange sky and distant, low hills. The overall color palette is dominated by shades of yellow and orange.

C++
算法

动态规划

dynamic programming

二维DP

整数拆分

搭楼梯1

作为建筑师，你要用 n 块砖头(高度为1)搭出高度正好为 m 的楼梯，高度从左到右不能降低。输入 n 和 m ， $m \leq n \leq 10$ ，请问共有几种方案？

输入样例

7 3

输出样例

4

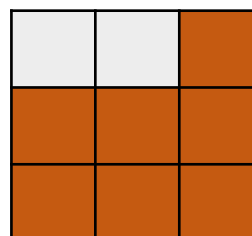
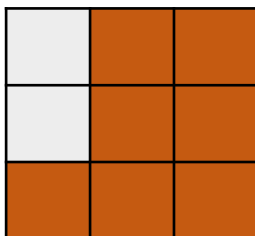
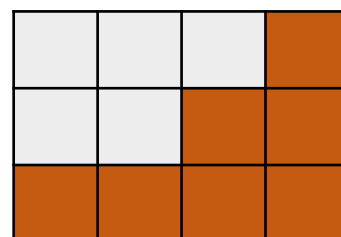
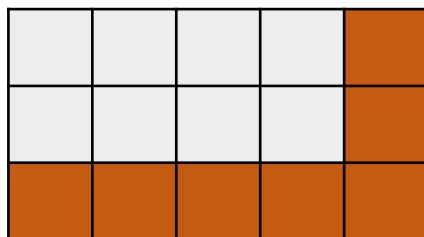
说明

$7 = 1 + 1 + 1 + 1 + 3$

$7 = 1 + 1 + 2 + 3$

$7 = 1 + 3 + 3$

$7 = 2 + 2 + 3$



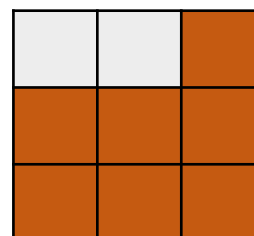
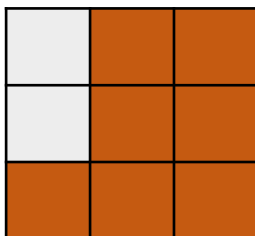
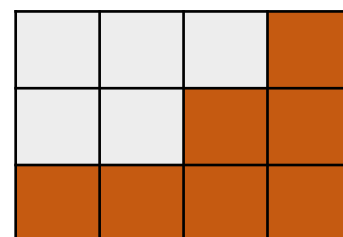
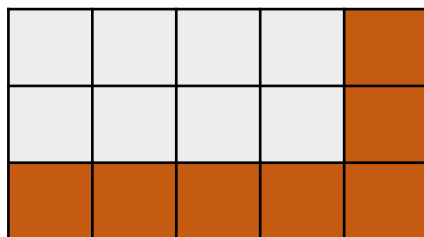
整数拆分1

作为数学家，你要把正整数 n 拆分成若干个正整数，其中最大数正好为 m 。
输入 n 和 m ， $m \leq n \leq 10$ ，请问共有几种方案？

输入样例
7 3

输出样例
4

说明
 $7=1+1+1+1+3$
 $7=1+1+2+3$
 $7=1+3+3$
 $7=2+2+3$



注意以下拆分等效，只能算一种： $7=1+3+3$ ， $7=3+1+3$ ， $7=3+3+1$

搭楼梯2

作为建筑师，你要用 n 块砖头(高度为1)搭出正好 m 个横向踏步的楼梯，高度从左到右不能上升。输入 n 和 m ， $m \leq n \leq 10$ ，请问共有几种方案？

输入样例

7 3

输出样例

4

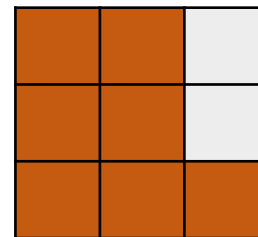
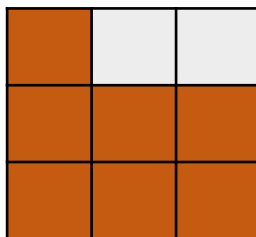
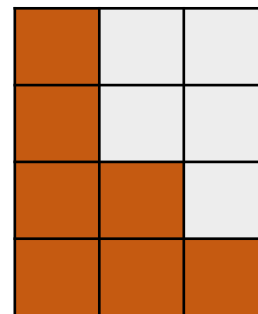
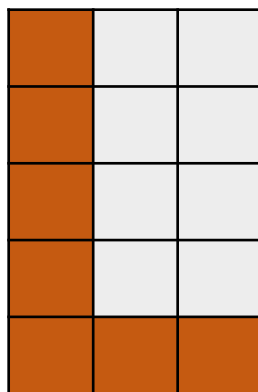
说明

$7=5+1+1$

$7=4+2+1$

$7=3+2+2$

$7=3+3+1$



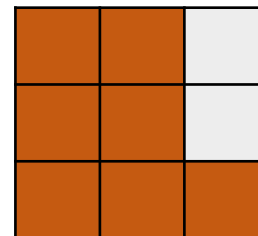
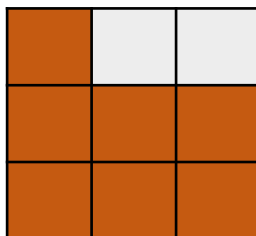
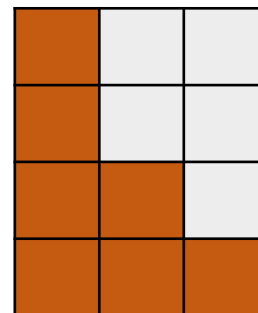
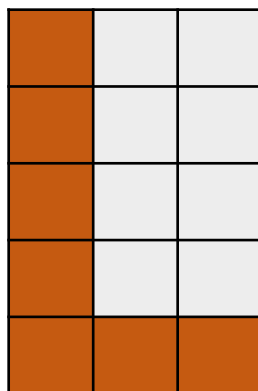
整数拆分2

作为数学家，你要把正整数 n 拆分成正好 m 个正整数。
输入 n 和 m ， $m \leq n \leq 10$ ，请问共有几种方案？

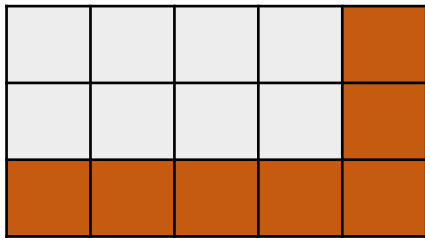
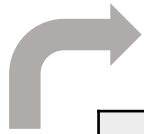
输入样例
7 3

输出样例
4

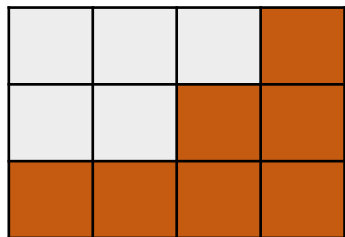
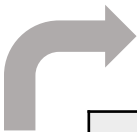
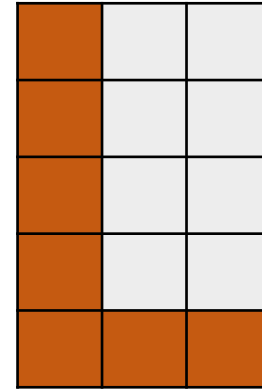
说明
 $7=5+1+1$
 $7=4+2+1$
 $7=3+2+2$
 $7=3+3+1$



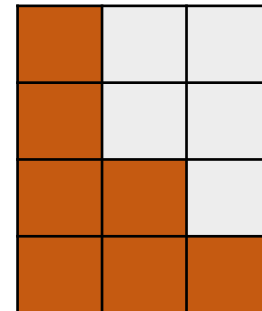
Ferrers图



顺时针
旋转



顺时针
旋转



两类问题的统一

正整数 n 拆分成
最大数正好为 m
的若干个正整数
的方案数

=

Ferrers图

正整数 n 拆分成
正好 m 个正整数
的方案数

n 的 m 拆分数

求解： 整数拆分

作为数学家，你要把正整数 n 拆分成若干个正整数，其中最大数正好为 m 。
输入 n 和 m ， $m \leq n \leq 10$ ， 请问共有几种方案？

输入样例
7 3

输出样例
4

说明
 $7=1+1+1+1+3$
 $7=1+1+2+3$
 $7=1+3+3$
 $7=2+2+3$

思考： 为什么是二维DP？

如何设计状态 i, j ？

如何设计 $f[i][j]$ ， 代表什么含义？

如何建立 $f[i][j]$ 和其他状态的联系？

求解： 整数拆分

作为数学家，你要把正整数 n 拆分成若干个正整数，其中最大数正好为 m 。
输入 n 和 m ， $m \leq n \leq 10$ ， 请问共有几种方案？

输入样例

7 3

输出样例

4

说明

$7=1+1+1+1+3$

$7=1+1+2+3$

$7=1+3+3$

$7=2+2+3$

最自然的设计：仿照原问题
因为原题"拆分 n ,最大为 m 的方案数"
所以 $f[i][j]$ 代表拆分 i ,最大为 j 的方案数

$f[i][j]$ 代表拆分 i ,最大为 j 的方案数

	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$
$i=0$	0	0	0	0	0
$i=1$	0	1	0	0	0
$i=2$	0	1	1	0	0
$i=3$	0	1	1	1	0
$i=4$	0	1	2	1	1
$i=5$	0	1	2	2	1
$i=6$	0	1	3	3	2
$i=7$	0	1	3	4	3
$i=8$	0	1	4	5	5
$i=9$	0	1	4	7	6

边界
条件

缓冲带

二维DP
核心步骤
就是
二维填表

缓冲带

动态
填表

$f[i][j]$ 代表拆分*i*,最大为*j*的方案数

$(i,j)=(1,1)$	$f[1][1] = 1$
$i < j$	$f[i][j] = 0$

边界条件

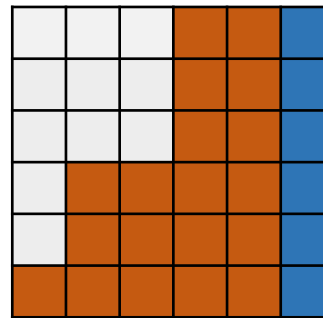
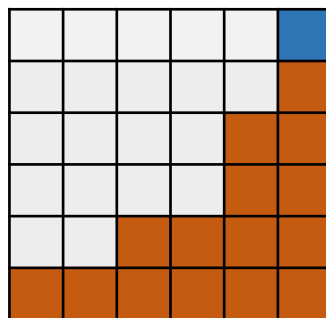
$(i,j) \neq (1,1)$ 并且 $i > j$	$f[i][j] = f[i-1][j-1] + f[i-j][j]$
----------------------------------	-------------------------------------

状态转移方程

2种情况

只有1个*j*

不止1个*j*



*j*变成*j*-1

将1个*j*清零

$ans = f[n][m]$

最终答案

对边界条件，可以有不同处理

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	0	0	0
i=2	0	1	1	0	0
i=3	0	1	1	1	0
i=4	0	1	2	1	1
i=5	0	1	2	2	1
i=6	0	1	3	3	2
i=7	0	1	3	4	3
i=8	0	1	4	5	5
i=9	0	1	4	7	6

当 $j > i$ 时,
 $f[i][j]=0$
无需再计算

$$f[8][3] = f[7][2] + f[5][3]$$

$$f[9][4] = f[8][3] + f[5][4]$$

整数拆分:代码1

填表顺序：从上往下，从左往右

```
5 int n,m,f[N][M];
6 int main(){
7     cin>>n>>m;
8     f[1][1]=1;      第1行第1列边界初始化
9     for(int i=2;i<=n;i++)
10         for(int j=1;j<=min(i,m);j++)  j<=i会错
11             f[i][j]=f[i-1][j-1]+f[i-j][j];
12     cout<<f[n][m]<<endl;
```

易错点 $i-j$ 越界

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	0	0	0
i=2	0	1	1	0	0
i=3	0	1	1	1	0
i=4	0	1	2	1	1
i=5	0	1	2	2	1
i=6	0	1	3	3	2
i=7	0	1	3	4	3
i=8	0	1	4	5	5
i=9	0	1	4	7	6

每一格
只依赖
左上角
和
正上方
元素

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	0	0	0
i=2	0	1	1	0	0
i=3	0	1	1	1	0
i=4	0	1	2	1	1
i=5	0	1	2	2	1
i=6	0	1	3	3	2
i=7	0	1	3	4	3
i=8	0	1	4	5	5
i=9	0	1	4	7	6

填表顺序
可以选择
从上往下
从左往右

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	0	0	0
i=2	0	1	1	0	0
i=3	0	1	1	1	0
i=4	0	1	2	1	1
i=5	0	1	2	2	1
i=6	0	1	3	3	2
i=7	0	1	3	4	3
i=8	0	1	4	5	5
i=9	0	1	4	7	6

填表顺序
可以选择
从左往右
从上往下

第1列：边界初始化

整数拆分:代码2

填表顺序：从左往右，从上往下

```
6 int n,m,f[N][M];
7 int main(){
8     cin>>n>>m;
9     for(int i=1;i<=n;i++)f[i][1]=1;
10    for(int j=2;j<=m;j++)
11        for(int i=j;i<=n;i++)
12            f[i][j]=f[i-1][j-1]+f[i-j][j];
```

第1列
边界初始化

填表顺序可以选择斜线：从上往下，从左上往右下

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	0	0	0
i=1	0	1	0	0	0
i=2	0	1	1	0	0
i=3	0	1	1	1	0
i=4	0	1	2	1	1
i=5	0	1	2	2	1
i=6	0	1	3	3	2
i=7	0	0	3	4	3
i=8	0	0	0	5	5
i=9	0	0	0	0	6

i 等于 j 时
边界初始化

填表顺序可以
选择斜线
从上往下
从左上往右下

当 $i-j > n-m$ 时, 无需再计算

整数拆分:代码3

填表顺序： 对角线斜向右下

```
6 int n,m,f[N][M];
```

```
7 int main(){
```

```
8     cin>>n>>m;
```

行数i等于列数j时，边界初始化

```
9     for(int j=1;j<=m;j++)f[j][j]=1;
```

```
10    for(int k=1;k<=n-m;k++)
```

k代表行数i减列数j

```
11        for(int j=1;j<=m;j++){
```

列数j扫描1到m

```
12            int i=j+k;
```

定位行数为i

```
13            f[i][j]=f[i-1][j-1]+f[i-j][j];
```

```
14        }
```

整数拆分:代码4

滚动数组
节省空间

填表顺序：从左往右，从上往下
其他填表顺序可以用滚动数组吗？

```
5  int n,m,f[N][2];
6  int main(){
7      cin>>n>>m;
8      for(int i=1;i<=n-m+1;i++)f[i][1]=1;
9      int r=0;
10     for(int j=2;j<=m;j++,r=1-r)
11         for(int i=j;i<=n-m+j;i++)
12             f[i][r]=f[i-1][1-r]+f[i-j][r];
13     cout<<f[n][1-r]<<endl;
```

第1列
边界初始化

r代表滚动数
组待填写列

整数拆分:代码4

滚动数组
节省空间

填表顺序：从左往右，从上往下
其他填表顺序可以用滚动数组吗？

```
5  int n,m,f[N][2];
6  int main(){
7      cin>>n>>m;
8      for(int i=1;i<=n-m+1;i++)f[i][1]=1;
9      int r=0;
10     for(int j=2;j<=m;j++,r=1-r)
11         for(int i=j;i<=n-m+j;i++)
12             f[i][r]=f[i-1][1-r]+f[i-j][r];
13     cout<<f[n][1-r]<<endl;
```

第1列
边界初始化

r代表滚动数
组待填写列

此代码错在哪里？

整数拆分方案计数问题汇总

0

n 拆分成若干个正整数，最大数正好为 m



1

n 拆分成正好 m 个正整数



2

n 拆分成若干个正整数，最大数不超过 m

枚举最大数 $m'=1,2,\dots,m$

转化为多个问题零



3

n 拆分成若干个正整数

枚举最大数 $m'=1,2,\dots,n$

转化为多个问题一



4

n 拆分成 m 个偶数

修改 m 个数:每个数除以2

转化为问题一 $f[n/2][m]$



整数拆分方案计数问题汇总

5

n 拆分成 m 个奇数

修改 m 个数:每个数先加1再除以2

转化为问题一 $f[(n+m)/2][m]$



6

n 拆分成 m 个不同的正整数

修改 m 个数:第 i 小数减 $(i-1)$

转化为问题一 $f[n-m*(m-1)/2][m]$



7

n 拆分成若干个奇数

枚举奇数个数 $m'=1,2,3,\dots$

转化为多个问题五的答案再求和



8

n 拆分成若干个不同的正整数

枚举不同数的个数 $m'=1,2,3,\dots$

转化为多个问题六的答案再求和



9

n 拆分成若干个不同的奇数

?

易错点：数据范围

思考：使用int,long long或unsigned long long类型
针对以下问题，可以求解多大的输入数据而不溢出？

n的阶乘

第n项斐波那契数列

第n项卡特兰数

组合数 $C(n,m)$

n的m拆分数

整数拆分 综合练习

课件下载链接:

链接: <https://pan.baidu.com/s/1ei7f7w>

密码: q66i

作业网站:

<http://120.132.18.213:8080/thrall-web/main#home>