



信奥算法

背包问题

knapsack problem

动态规划

dynamic programming



01背包：可行性

有一个背包最多载重 m 斤,共有 n 件物品,第 i 件物品重量为 $w[i]$ 。
对每件物品可选择拿走或不拿。请问能否恰好拿到总重量为 m 斤?

输入正整数 m 和 n ,再输入 n 个正整数代表物品重量。 $n \leq 100, m \leq 200$

输入样例:

2 10
3 8

输出样例:

No

输入样例:

3 6
2 1 3

输出样例:

Yes

| | | |
|----------|----|---|
| 01 决策 | 不选 | 0 |
| | 选 | 1 |



走楼梯：可行性

共 m 级楼梯, 有 n 种走法, 第 i 种走法一次往上走 $x[i]$ 级。对每种走法最多可以选用1次。请问能否恰好走到第 m 级?

输入正整数 m 和 n , 再输入 n 种走法对应级数。 $n \leq 100, m \leq 200$

输入样例:

2 10
3 8

输出样例:

No

输入样例:

3 6
2 1 3

输出样例:

Yes

01

决策

不选

0

选

1



凑数：可行性

目标凑出数字 m , 有 n 个数字可以使用, 第 i 个数为 $x[i]$ 。对每个数最多可以选用**1**次。请问能否恰好凑出 m ?

输入正整数 m 和 n , 再输入 n 个数字。 $n \leq 100, m \leq 200$

输入样例:

2 10
3 8

输出样例:

No

输入样例:

3 6
2 1 3

输出样例:

Yes

| | | |
|----------|----|---|
| 01 决策 | 不选 | 0 |
| | 选 | 1 |



简化问题

简化 n

先解决 $n=1$ 情况

再解决 $n=2$ 情况

再解决 $n=3$ 情况

.....

简化 m

先解决 $m=1$ 情况

再解决 $m=2$ 情况

再解决 $m=3$ 情况

.....

答案

$f[n][m]$ 表示用 n 个数字能否凑出 m

状态

$f[i][j]$ 表示只用前 i 个数字能否凑出 j

抄原
题大
法好

$f[i][j]$ 表示只用前*i*个数字能否凑出*j*

$m=7$

$n=3$

$x[1]=2$

$x[2]=4$

$x[3]=3$

| | $j=0$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=5$ | $j=6$ | $j=7$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $i=0$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $i=1$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $i=2$ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $i=3$ | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

$f[3][7]=f[2][7] \text{ or } f[2][4]$

决策

不选用
 $x[3]$

决策

选用
 $x[3]$

```
1 /*姓名XXX
2 输入n=3,m=7
```

电脑上完成此页注释

```
3 x[1]=2
```

3分钟后老师检查

```
4 x[2]=4
```

```
5 x[3]=3
```

```
6 f[i][j]表示只用前i个数字能否凑出j
```

```
7     j=0,j=1,j=2,j=3,j=4,j=5,j=6,j=7
```

```
8 i=0    1    0    0    0    0    0    0    0
```

```
9 i=1    1    0    1    0    0    0    0    0
```

```
10 i=2    1    0    1    0    1    0    1    0
```

```
11 i=3    1    0    1    1    1    1    1    1
```

```
12 */
```


输入正整数m和n，再输入n个数字。

$n \leq 100, m \leq 200$

```
13 #include<bits/stdc++.h>
14 using namespace std;
15 const int N=
16 const int M=
17 int n,m,x[N],f[N][M];
```

```
18 int main(){
19     cin>>n>>m;
20     for(int i=1;i<=n;++i) cin>>x[i];
21     f[0][0]=1;
22     for(int i=1;i<=n;++i)
23         for(int j=0;j<=m;++j)
24             if(j<x[i])
25                 f[i][j]=f[i-1][j];
26             else
27                 f[i][j]=f[i-1][j] or f[i-1][j-x[i]];
28     if(f[n][m])cout<<"Yes"<<endl;
29     else cout<<"No"<<endl;
30     return 0;
31 }
```

01背包：可行性

$f[i][j]$ 表示只用前 i 个数字能否凑出 j

初始
条件

$$f[0][0]=1$$

状态
转移
方程

若 $j < x[i]$

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

若 $j \geq x[i]$

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

01背包： 3种问题

可行性
判定问题

用 n 个物品能否恰好凑出 m 斤重量

方案
计数问题

用 n 个物品恰好凑出 m 斤共几种方案

最优化
问题

用 n 个物品凑出不超过 m 斤时最多几斤

凑数：3种问题

可行性
判定问题

用 n 个数字能否恰好凑出 m

方案
计数问题

用 n 个数字恰好凑出 m 共几种方案

最优化
问题

用 n 个数字凑出不超过 m 的最大数是几

01背包： 方案计数

有一个背包最多载重 m 斤,共有 n 件物品,第 i 件物品重量为 $w[i]$ 。
对每件物品可选择拿走或不拿。请问恰好凑出 m 斤共几种方案?

输入正整数 m 和 n , 再输入 n 个正整数代表物品重量。 $n \leq 100, m \leq 200$

输入样例:

2 10
3 8

输出样例:

0

输入样例:

4 6
2 1 5 3

输出样例:

2

$$6 = 2 + 1 + 3$$

$$6 = 1 + 5$$

方案与排列
顺序无关

| | | |
|----------|----|---|
| 01 决策 | 不选 | 0 |
| | 选 | 1 |



01背包： 方案计数

有一个背包最多载重 m 斤,共有 n 件物品,第 i 件物品重量为 $w[i]$ 。
对每件物品可选择拿走或不拿。请问恰好凑出 m 斤共几种方案?

输入正整数 m 和 n ,再输入 n 个正整数代表物品重量。 $n \leq 100, m \leq 200$

状态
定义

抄原题
大法好

答案

$g[n][m]$ 表示用 n 个数字恰好凑出 m 共几种方案

状态

$g[i][j]$ 表示用前 i 个数字恰好凑出 j 共几种方案

$g[i][j]$ 表示用前*i*个数字恰好凑出*j*共几种方案

$m=6$

$n=4$

$x[1]=2$

$x[2]=5$

$x[3]=1$

$x[4]=3$

| | $j=0$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=5$ | $j=6$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| $i=0$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $i=1$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| $i=2$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| $i=3$ | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| $i=4$ | 1 | 1 | 1 | 2 | 1 | 2 | 2 |

$$g[4][6] = g[3][6] + g[3][6-3]$$

决策

不选用
 $x[4]$

决策

选用
 $x[4]$

```
1 /*姓名XXX
2 输入n=4,m=6
3 x[1]=2
4 x[2]=5
5 x[3]=1
6 x[4]=3
```

电脑上完成此页注释

3分钟后老师检查

```
7 g[i][j]表示用前i个数字恰好凑出j共几种方案
8   j=0,j=1,j=2,j=3,j=4,j=5,j=6
```

```
9 i=0   1   0   0   0   0   0   0
```

```
10 i=1   1   0   1   0   0   0   0
```

```
11 i=2   1   0   1   0   0   1   0
```

```
12 i=3   1   1   1   1   0   1   0
```

```
13 i=4   1   1   1   2   1   2   2
```

```
14 */
```

01背包： 方案计数

$g[i][j]$ 表示用前 i 个数字恰好凑出 j 共几种方案

初始
条件

$$g[0][0]=1$$

状态
转移
方程

若 $j < x[i]$

$$g[i][j]=g[i-1][j]$$

若 $j \geq x[i]$

$$g[i][j]=g[i-1][j] + g[i-1][j-x[i]]$$

```
20 int main(){
21     cin>>n>>m;
22     for(int i=1;i<=n;++i) cin>>x[i];
23     
24     for(int i=1;i<=n;++i)
25         for(int j=0;j<=m;++j)
26             if(j<x[i])
27                 g[i][j]=g[i-1][j];
28             else
29                 
30     cout<<g[n][m]<<endl;
31     return 0;
32 }
```

01背包：最优化问题

有一个背包最多载重 m 斤,共有 n 件物品,第 i 件物品重量为 $w[i]$ 。
对每件物品可选择拿走或不拿。请问用这个包最多拿走几斤物品?

输入正整数 m 和 n ,再输入 n 个正整数代表物品重量。 $n \leq 100, m \leq 200$

输入样例:

2 10
3 8

输出样例:

8

输入样例:

3 7
4 1 5

输出样例:

6

| | | |
|----------|----|---|
| 01 决策 | 不选 | 0 |
| | 选 | 1 |



01背包：最优化问题

有一个背包最多载重 m 斤,共有 n 件物品,第 i 件物品重量为 $w[i]$ 。
对每件物品可选择拿走或不拿。请问用这个包最多拿走几斤物品?

输入正整数 m 和 n ,再输入 n 个正整数代表物品重量。 $n \leq 100, m \leq 200$

状态
定义

抄原题
大法好

答案

$h[n][m]$ 表示用 n 个物品载重 m 的包最多拿几斤

状态

$h[i][j]$ 表示用 i 个物品载重 j 的包最多拿几斤

$h[i][j]$ 表示用*i*个物品载重*j*的包最多拿几斤

$m=6$

$n=4$

$x[1]=4$

$x[2]=1$

$x[3]=5$

| | $j=0$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=5$ | $j=6$ | $j=7$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $i=0$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $i=1$ | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| $i=2$ | 0 | 1 | 1 | 1 | 4 | 5 | 5 | 5 |
| $i=3$ | 0 | 1 | 1 | 1 | 4 | 5 | 6 | 6 |

$$h[3][7] = \max(h[2][7], h[2][7-5] + 5)$$

决策

不选用
 $x[3]$

决策

选用
 $x[3]$

01背包：最优化问题

$h[i][j]$ 表示用 i 个物品载重 j 的包最多拿几斤

初始
条件

$$h[0][0]=1$$


状态
转移
方程

若 $j < x[i]$

$$h[i][j]=h[i-1][j]$$

若 $j \geq x[i]$

$$h[i][j]=\max(h[i-1][j], h[i-1][j-x[i]]+x[i])$$

```
21 cin>>n>>m;  
22 for(int i=1;i<=n;++i) cin>>x[i];  
23 for(int i=1;i<=n;++i)  
24     for(int j=0;j<=m;++j)  
25         if(j<x[i])  
26             h[i][j]=h[i-1][j];  
27         else  
28               
29 cout<<h[n][m]<<endl;
```

作业要求

写程序前请写明：

1. 二维数组每一格的含义
2. 手算样例对应表格

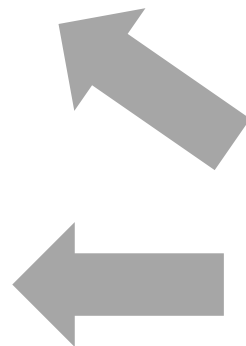
查错方法：

1. 打印二维数组
2. 和手算表格对比找不同

```
1  /*
2  f[i][j]代表
3      j=0,1,2,3,4,5
4  i=0
5  i=1
6  i=2
7  i=3
8  i=4
9  i=5
10 */
```



模版格式
供参考



调试查错方法



```
30 // 查错代码
31 // for(int i=1;i<=n;++i,cout<<endl)
32 //     for(int j=0;j<=m;++j)cout<<h[i][j]<<" ";
```

快快编程作业

| | |
|------|---------|
| 1905 | 参考可行性问题 |
|------|---------|

| | |
|------|----------|
| 1906 | 参考方案计数问题 |
|------|----------|

| |
|-----|
| 322 |
|-----|

| | |
|-----|--------------------------|
| 拓展题 | 5, 141, 246, 252, 8, 110 |
|-----|--------------------------|