

CS101

A Mars rover, likely a Curiosity rover, is shown on a rocky, orange-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 sun is bright in the upper right corner, casting a long shadow of the rover onto the ground.

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>

高级递推算法

动态规划

大盗：两种金砖



金块A, 3斤



金块B, 8斤

金块很多，
该怎么选？



只能装10斤

大盗：两种金砖， 足够多

大盗去抢银行，他带的包只能装 n 公斤的金块。现在银行只有两种规格的金块，一种重 a 公斤，另一种重 b 公斤，数量足够多。输入 n, a, b ，输出他最多带走多少公斤。

输入样例：
10 3 8

输入样例：
10 11 12

输入样例：
8 4 3

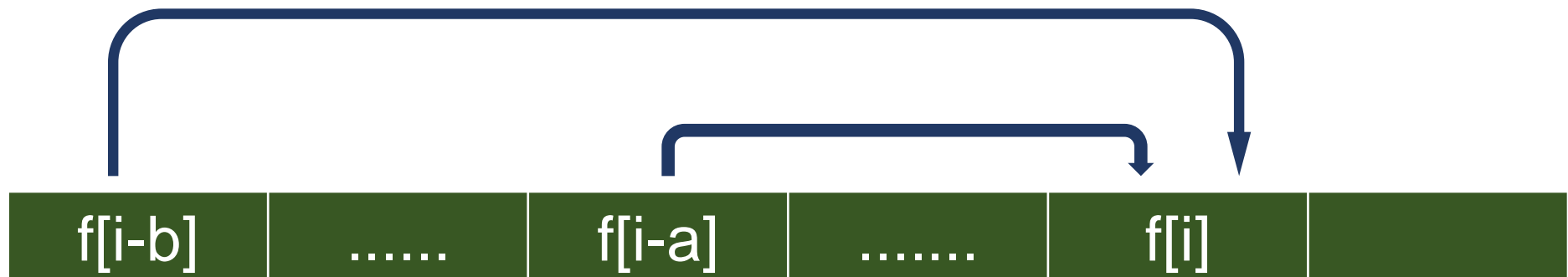
输出样例：
9

输出样例：
0

输出样例：
8

大盗：两种金砖， 足够多

$f[i]$ 表示用 i 公斤以内的包最多拿多少金块



大盗：两种金砖， 足够多

$f[i]$ 表示用*i*公斤以内的包最多拿多少金块

当*i*为0时

$$f[0] = 0$$

当*i*<*a*并且*i*<*b*时

$$f[i] = 0$$

当*i*>=*a*并且*i*<*b*时

$$f[i] = f[i - a] + a$$

当*i*<*a*并且*i*>=*b*时

$$f[i] = f[i - a] + b$$

当*i*>=*a*并且*i*>=*b*时

$$f[i] = \max(f[i - a] + a, f[i - b] + b)$$

大盗：两种金砖， 足够多

$f[i]$ 表示用*i*公斤以内的包最多拿多少金块

当*i*为0时

$$f[0] = 0$$

初始时*f*值
都清零

$$f[i] = \max(\begin{array}{l} f[i - a] + a | i \geq a, \\ f[i - b] + b | i \geq b \end{array})$$

若*i*≥*a*，可以拿重*a*的金块

若*i*≥*b*，可以拿重*b*的金块


```
1 #include<iostream>
2 #define M 100001
3 using namespace std;
4 int f[M],n,a,b,xa,xb,i,j;
5 int main(){
6     cin>>n>>a>>b;
7     f[0]=0;
8     for(i=1;i<=n;i++){
9         if(i>=a) xa=f[i-a]+a;
10        else xa=0;
11        if(i>=b) xb=f[i-b]+b;
12        else xb=0;
13        f[i]=max(xa,xb);
14    }
15    cout<<f[n]<<endl;
16    return 0;
17 }
```

银行盗贼：三种金砖，足够多

盗贼去抢银行，他带的包只能装n公斤的金块。现在银行只有三种规格的金块，分别重a公斤，b公斤和c公斤，数量足够多，金块无法分割。输入整数n,a,b,c，输出他最多带走多少公斤。

$f[i]$ 表示用i公斤以内的包最多拿多少金块

当i为0时

$$f[0] = 0$$

$$f[i] = \max(\begin{aligned} &f[i - a] + a \mid i \geq a, \\ &f[i - b] + b \mid i \geq b, \\ &f[i - c] + c \mid i \geq c \end{aligned})$$

若 $i \geq a$

若 $i \geq b$

若 $i \geq c$

初始时f值
都清零

银行盗贼：三种金砖， 足够多

$f[i]$ 表示用 i 公斤以内的包最多拿多少金块



大盗的烦恼：m种金砖，足够多

盗贼去抢银行，他带的包只能装n公斤的金块。
银行有m种规格的金块，分别重 x_1, x_2, \dots, x_m 公斤，
数量足够多，金块无法分割。
输入整数n, m, x_1, x_2, \dots, x_m ，输出最多带走多少公斤

$f[i]$ 表示用i公斤以内的包最多拿多少金块

当i为0时

$$f[0] = 0$$

$$f[i] = \max_{j: x_j \leq i} \{ f[i - x_j] + x_j \}$$

初始时f值
都清零

在所有满足 $x_j \leq i$ 的金
块j中选择,求最优决策

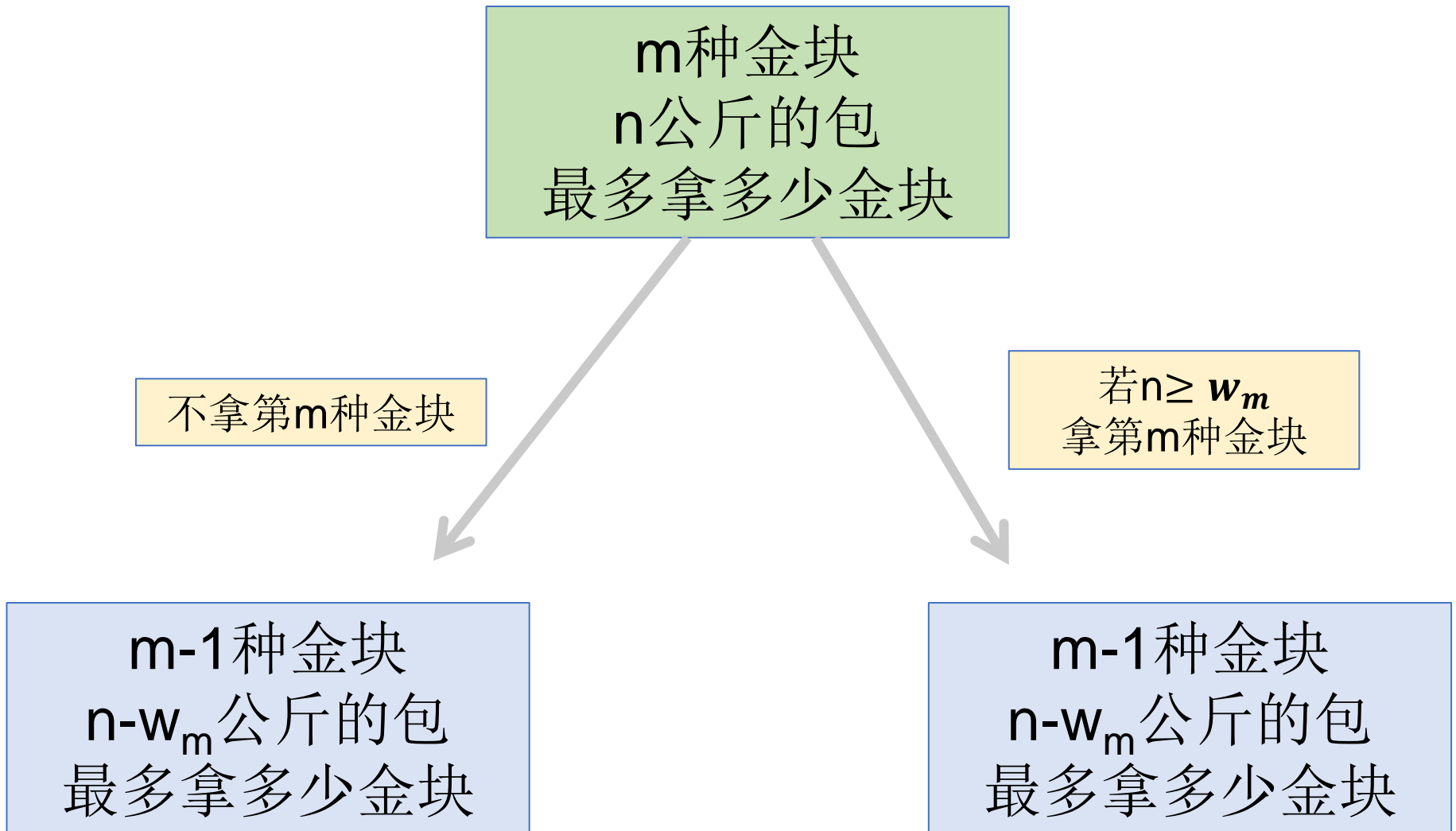
大盗： m 种金砖，01限制

盗贼去抢银行，他带的包只能装 n 公斤的金块。
银行有 m 种规格的金块，分别重 w_1, w_2, \dots, w_m 公斤，
数量都**只有1块**，金块无法分割。输入整数 $n, m,$
 w_1, w_2, \dots, w_m ，输出**最多带走多少公斤**

对于第 i 种金块，两个可能决策：
不拿第 i 种金块；拿第 i 种金块

枚举算法：所有可能性共 2^m 种可能

原问题 分解成 子问题



大盗：m种金砖，01限制

盗贼去抢银行，他带的包只能装 n 公斤的金块。银行有 m 种规格的金块，分别重 w_1, w_2, \dots, w_m 公斤，数量都只有1块，金块无法分割。

输入整数 $n, m, w_1, w_2, \dots, w_m$ ，输出**最多带走多少公斤**

递
推
算
法

二维递推算法： $f[i][j]$ 表示只装前 i 种金块，用 j 公斤以内的包，最多拿多少金块

当 i 为0时 $f[0][j] = 0$

当 j 为0时 $f[i][0] = 0$

大盗：m种金砖，01限制

二维递推算法：

$f[i][j]$ 表示只装前*i*种金块，
用*j*公斤以内的包，最多拿多少金块

当*i*为0时 $f[0][j] = 0$

当*j*为0时 $f[i][0] = 0$

$$f[i][j] = \max($$
$$f[i-1][j],$$
$$f[i-1][j-w_i] + w_i \mid j \geq w_i$$
$$)$$

若 $j \geq w_i$
可以拿第*i*件

承重
 $j-w_i$

承重
 j

第*i*-1种金砖

第*i*种金砖

...	...	$f[i-1][j-w_i]$	$f[i-1][j]$
					$f[i][j]$

大盗m种金砖01限制- 代码

```
1  #include <iostream>
2  #define M 505
3  #define N 2005
4  using namespace std;
5  int n,m,w[M],v[M],f[M][N];
6  int main(){
7      cin>>n>>m;
8      for(int i=1;i<=m;i++) cin>>w[i];
9      //初始化清零, 已经默认执行了
10     for(int i=1;i<=m;i++) //循环查看每种金块i
11         for(int j=0;j<=n;j++) { //循环查看背包剩余承重j
12             if(j<w[i]) //金块i太重, 无法放入
13                 f[i][j]=f[i-1][j];
14             else //比较两种决策: 金块i可以放, 或者不放
15                 f[i][j]=max(f[i-1][j],f[i-1][j-w[i]]+w[i]);
16         }
17     cout<<f[m][n]<<endl;
18     return 0;
19 }
```

讨论题

为什么递推算法比暴力枚举算法快？

递推时储存已
计算完的结果

计算次数约
 $O(n*m)$

暴力枚举时重复
计算了子问题

计算次数约
 $O(2^m)$