



# 信奥算法

# 泛化背包

generalized knapsack

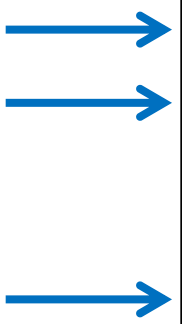
# 分组背包



# 分组01背包

背包最大载重 $C$ ,有 $n$ 件物品,重量分别是 $w_1, w_2, \dots, w_n$ , 价值分别为 $v_1, v_2, \dots, v_n$ 。这些物品被划分为若干组, **每组中的物品互相冲突,且最多选一件**。选出的物品的总重量不能超过 $C$ ,求总价值最大。

输入第1行3个整数,  $C$ (背包载重,  $0 \leq C \leq 200$ ),  $n$ (物品数量,  $1 \leq n \leq 30$ )和  $G$ (最大组号,  $0 \leq G \leq 10$ ); 接下来 $n$ 行,每行三个整数 $w_i, v_i, p$ ,表示每个物品的重量,价值和所属分组。



输入样例	输出样例
10 6 3	20
2 1 1	
3 3 1	
4 8 2	
6 9 2	
2 8 3	
3 9 3	

载重10,共6件物品,分组号{0,1,2,3}
重量2,价值1,所属组号1
重量3,价值3,所属组号1
重量4,价值8,所属组号2
重量6,价值9,所属组号2
重量2,价值8,所属组号3
重量3,价值9,所属组号3

# 分组01背包

定义状态：用抄原题大法尝试

$f[i][j]$ 表示只装前*i*组物品  
用载重*j*的包最多拿多少价值

# 分组01背包

$f[i][j]$ 表示只装前*i*组物品  
用载重*j*的包最多拿多少价值

$$f[i][j] = \max\{$$

第*i*组全不要

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

枚举第*i*组里  
所有物品  
选哪一件

$$\max_{k \in G_i} \{ f[i-1][j - w_{ik}] + v_{ik} \mid j \geq w_{ik} \}$$

状态

[*i*]维度能否隐藏

# 分组01背包

$f[i][j]$	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=0$						
$i=1$						
$i=2$						

决策：第 $i$ 组选哪个物品

决策：第 $i$ 组不选任何物品

# 储存每一组的物品

```
3 typedef long long ll;  
4 const ll G=19;  
5 const ll N=39;  
6 const ll MAXV=209;  
7 ll c,n,g,f[MAXV];  
8 vector<ll> w[G],v[G];
```

w[p]储存p号组内物品重量

v[p]储存p号组内物品价值

```
10 cin>>c>>n>>g;  
11 for(ll i=1;i<=n;i++){  
12     ll ww,vv,p;  
13     cin>>ww>>vv>>p;  
14     w[p].push_back(ww);  
15     v[p].push_back(vv);  
16 }
```



# 分组01背包

```
17 for(11 i=0;i<=g;i++)
18     for(11 j=c;j>=0;j--)
19         for(11 k=0;k<w[i].size();k++)
20             if(j>=w[i][k])
21                 f[j]=max(f[j],f[j-w[i][k]]+v[i][k]);
22 cout<<f[c]<<endl;
```

从大到小  
枚举载重j



# 分组1背包

背包最大载重 $C$ ,有 $n$ 件物品,重量分别是 $w_1, w_2, \dots, w_n$ , 价值分别为 $v_1, v_2, \dots, v_n$ 。这些物品被划分为若干组, **每组中的物品恰要选出一件**。选出的物品的总重量不能超过 $C$ ,求总价值最大。

输入第1行3个整数,  $C$ (背包载重,  $0 \leq C \leq 200$ ),  $n$ (物品数量,  $1 \leq n \leq 30$ )和  $G$ (最大组号,  $0 \leq G \leq 10$ ); 接下来 $n$ 行,每行三个整数 $w_i, v_i, p$ ,表示每个物品的重量,价值和所属分组。

输入样例

10 6 3

2 1 1

4 3 1

4 8 2

9 20 2

2 8 3

3 9 3

输出样例

19

载重10,共6件物品,分组号{0,1,2,3}

重量2,价值1,所属组号1

重量3,价值3,所属组号1

重量4,价值8,所属组号2

重量6,价值9,所属组号2

重量2,价值8,所属组号3

重量3,价值9,所属组号3

定义  
状态

# 分组<sup>1</sup>背包

自然状态

A  $h[i][j]$ 表示前 $i$ 组恰最多用 $j$ 载重最多拿多少价值

具体化状态

B  $f[i][j]$ 表示前 $i$ 组恰好用 $j$ 载重最多拿多少价值

请问以上哪种状态  
可以解出此题

快快编程  
kkcoding.net

# 分组1背包

h[i][j]表示前i组恰最多用j载重最多拿多少价值

```

1  /*
2  h[i][j]表示前i组恰最多用j载重最多拿多少价值
3  c=4,n=3,g=2
4  w[1]=2,v[1]=5,p[1]=1;
5  w[2]=3,v[2]=1,p[2]=2;
6  w[3]=1,v[3]=2,p[3]=1;
7      j=0, j=1, j=2, j=3, j=4
8  i=0    0    0    0    0    0
9  i=1  -INF  2    5    5    5
10 i=2  
11 */

```

## 分组1背包

$h[i][j]$ 表示前 $i$ 组恰最多用 $j$ 载重最多拿多少价值

```
for(11 i=1;i<=g;i++){  
    h[i][0]=-INF; ←  
    for(11 j=1;j<=c;++j){  
        h[i][j]=h[i][j-1]; ←  
        for(11 k=0;k<w[i].size();k++){  
            if(j>=w[i][k]&&h[i-1][j-w[i][k]]>=0) ←  
                h[i][j]=max(h[i][j],h[i-1][j-w[i][k]]+v[i][k]);  
        }  
    }  
}
```

答案在 $h[?][?]$

# 分组1背包

$f[i][j]$ 表示前 $i$ 组恰好用 $j$ 载重最多拿多少价值

```

1  /*
2   $f[i][j]$ 表示前 $i$ 组恰好用 $j$ 载重最多拿多少价值
3   $c=4, n=3, g=2$ 
4   $w[1]=2, v[1]=5, p[1]=1;$ 
5   $w[2]=3, v[2]=1, p[2]=2;$ 
6   $w[3]=1, v[3]=2, p[3]=1;$ 
7       $j=0, j=1, j=2, j=3, j=4$ 
8   $i=0$      $0$      $-INF$   $-INF$   $-INF$   $-INF$ 
9   $i=1$ 

|  |  |  |  |  |  |
|--|--|--|--|--|--|
|  |  |  |  |  |  |
|--|--|--|--|--|--|


10  $i=2$      $-INF$   $-INF$   $-INF$   $-INF$   $3$ 
11 */

```

# 分组1背包

$f[i][j]$ 表示前 $i$ 组恰好用 $j$ 载重最多拿多少价值

```
for(int j=1;j<=c;++j)f[0][j]=-INF; ←  
for(int i=1;i<=g;i++)  
    for(int j=0;j<=c;++j){  
        f[i][j]=-INF; ←  
        for(int k=0;k<w[i].size();k++)  
            if(j<=w[i][k]&&f[i-1][j-w[i][k]]>=0) ←  
                f[i][j]=max(f[i][j],f[i-1][j-w[i][k]]+v[i][k]);  
    }
```

答案在 $f[?][?]$

# 分组1背包

请写出暴力dfs()函数的定义句  
包含几个参数?

```
void dfs(ll gid,ll weight,ll value)
```


搜索  
状态  
描述

当前考虑第gid组  
已用去载重weight  
已拿到价值value



# 分组1背包

```
10 void dfs(ll gid, ll weight, ll value){  
11  
12  
13  
14  
15 }
```

```
26 ans=-INF;  全局变量  
27 dfs(1,0,0);  
28 if(ans>0)cout<<ans<<endl;  
29 else cout<<-1<<endl;
```

泛化物品

广义物品

generalized item

快快编程  
kkcoding.net

## 泛化物品 收益函数

给花施肥1份	美观度0
给花施肥2份	美观度2
给花施肥3份	美观度5
给花施肥4份	美观度1

"花"这个物品  
"施肥"这件事  
理解为一种  
广义的"物品"  
分配若干"水"资源  
收益美观度

## 泛化物品 收益函数

张三吃 <b>0</b> 颗糖	幸福值 <b>0</b>
张三吃 <b>1</b> 颗糖	幸福值 <b>2</b>
张三吃 <b>2</b> 颗糖	幸福值 <b>5</b>
张三吃 <b>3</b> 颗糖	幸福值 <b>1</b>

"张三"这个人  
"吃糖"这件事  
理解为一种  
广义的"物品"  
分配若干"糖"资源  
收益幸福值

快快编程1936

快快编程  
kkcoding.net

$n$ 个人,有 $m$ 条亲戚关系(可传递),分成若干亲戚团体,每人有体重 $w[i]$ ,智商 $b[i]$ 。选若干人总体重不超过 $W$ , 要求:每个亲戚团可以全选/全不选/只选1人, 求总智商最高是多少。

请同学识别:哪些要素对应"分组01背包"?

背包  
要素

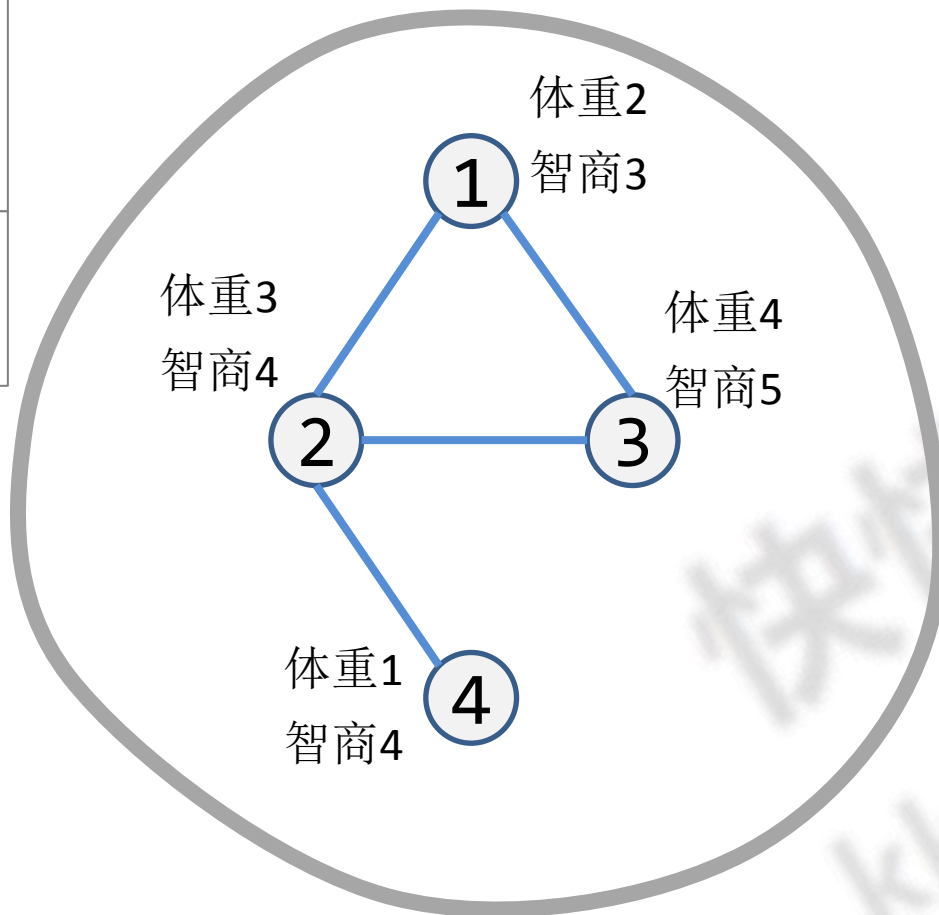
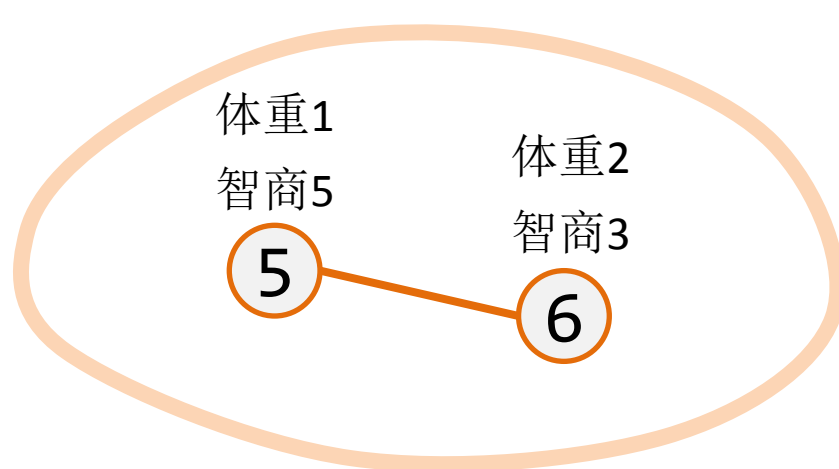
总体重不超过 $W$

总智商最高是多少

分组  
要素

每个亲戚团选人消耗体重指标  
每个亲戚团收益智商

输入
6 5 7
2 3 4 1 1 2
3 4 5 4 5 3
1 2
2 3
3 1
2 4
5 6
输出几



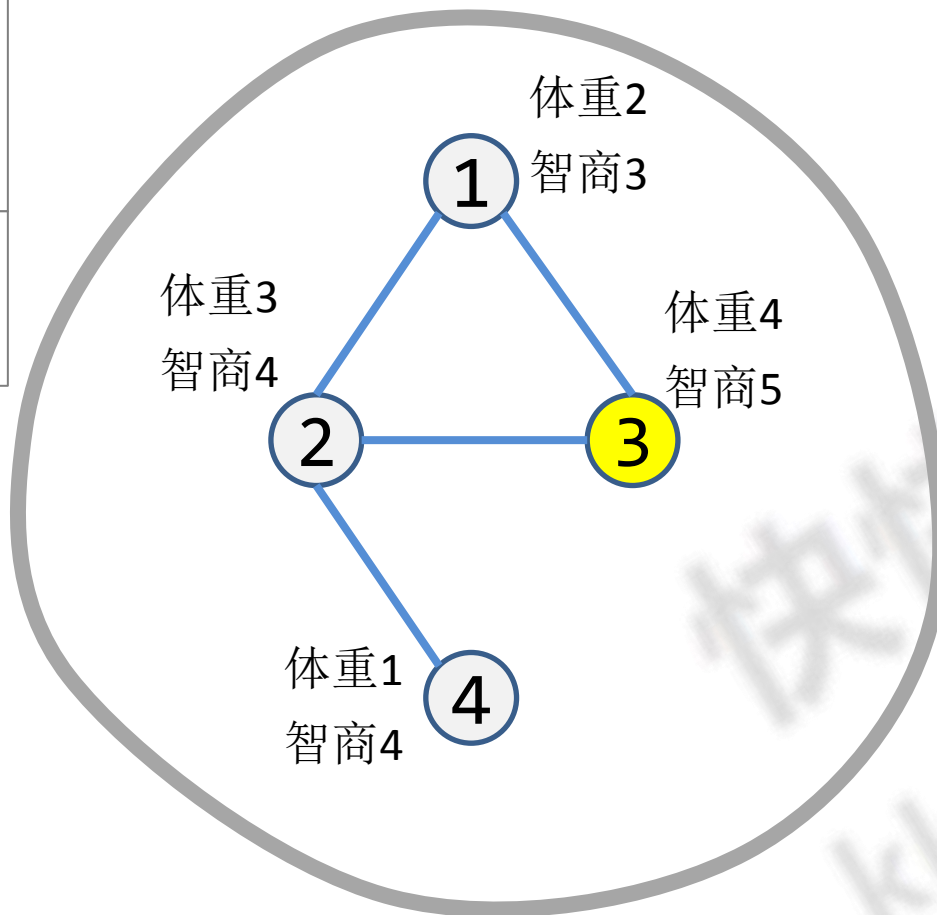
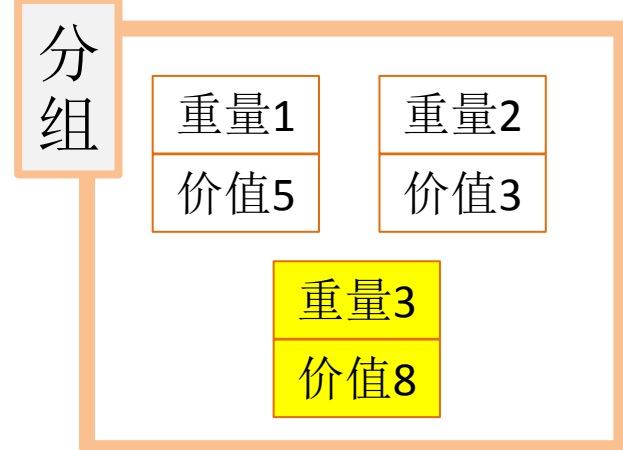
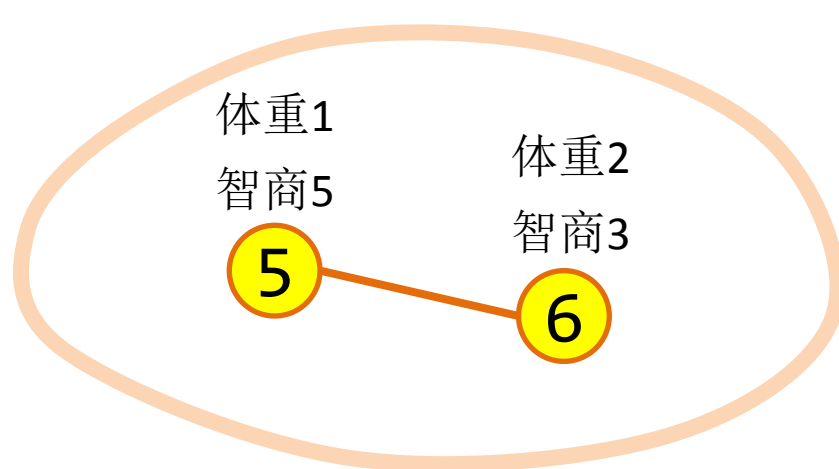


输入

6	5	7			
2	3	4	1	1	2
3	4	5	4	5	3
1	2				
2	3				
3	1				
2	4				
5	6				

输出

13
----



每个物品分组 等效于 1个泛化物品

每个分组内有  
{重量,价值}  
多种子物品

每个泛化物品  
消耗特定重量  
得到特定价值

每个分组内  
最多取1件物品

每个泛化物品  
选一种重量

分组01背包

泛化背包

分  
组

重量1

价值5

重量2

价值3

重量3

价值8

分  
组

重量2

价值3

重量3

价值4

重量4

价值5

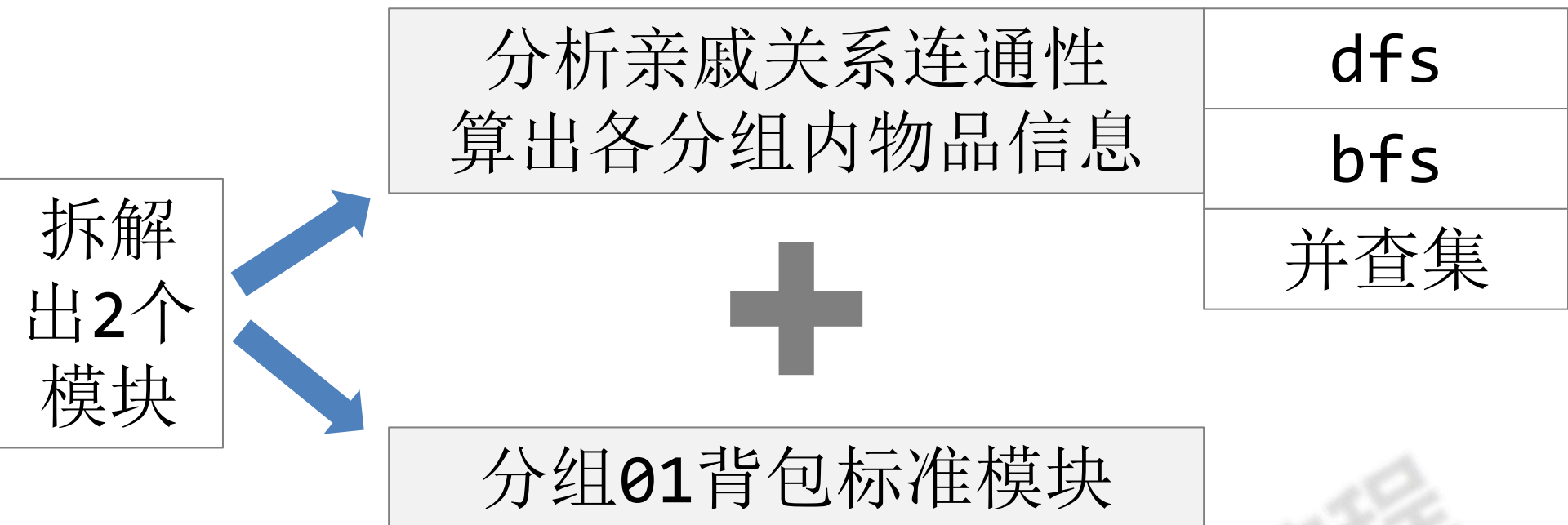
重量1

价值4

重量10

价值16

# 算法建模



```
8 int root(int x){
```

```
9
```

```
10 }
```

```
11 void unite(int x,int y){
```

```
12
```

```
13 }
```

```
15 cin>>n>>m>>W;
```

```
16 for(int i=1;i<=n;++i)cin>>w[i];
```

```
17 for(int i=1;i<=n;++i)cin>>b[i];
```

```
18
```

```
19 int x,y;
```

```
20 for(int i=1;i<=m;++i){
```

```
21     cin>>x>>y;
```

```
22     unite(x,y);
```

```
23 }
```

补全  
程序

(item){w,b}表示某亲戚团消耗体重w可以带来智商b

```
6 struct item{int w,b};
```

```
7 vector<item> g[N];
```

g[i][]储存i号为组长的小组里各个物品的信息

```
24 for(int i=1;i<=n;++i){  
25     sw[root(i)]+=w[i];  
26     sb[root(i)]+=b[i];  
27 }
```

root(i)能否  
改成id[i]

```
28 for(int i=1;i<=n;++i)  
29     g[root(i)].push_back((item){w[i],b[i]});
```

请写出第30,31行

```
30 for(int i=1;i<=n;++i)if(id[i]==i)  
31     g[i].push_back((item){sw[i],sb[i]});
```

```
6 struct item{int w,b;};  
7 vector<item> g[N];
```

`g[i][]`储存*i*号为组长的小组里各个物品的信息

```
for(int i=1;i<=n;++i)  
    for(int j=W;j>=0;--j)  
        for(int k=0;k<g[i].size();++k)  
            f[j]=  
cout<<f[W]<<endl;
```

补全  
程序

快快编程1937

快快编程  
kkcoding.net



n层书架,第i层c[i]本书,第i层第j本书价值v[i][j]. 只可以从每层两端取书,总数不超过m本,最多可以拿多少价值?

请同学识别:哪些要素对应"分组01背包"?

背包  
要素

总数量不超过m  
最多可以拿多少价值

分组  
要素

每层分配书籍数量  
每层收益价值

# 算法建模

$g[i][w]$ 表示第 $i$ 层拿最多 $w$ 本时  
该层最多拿多少价值

对每层预处理什么信息?

拆解  
出2个  
模块



分组01背包标准模块

预处理  
每一层

$g[i][w]$ 表示第 $i$ 层拿最多 $w$ 本时  
该层最多拿多少价值

举例：计算 $g[i][3]$   
第 $i$ 层共8本书

能否贪心？

请写出反例

预处理  
每一层

$g[i][w]$ 表示第*i*层拿最多*w*本时  
该层最多拿多少价值

举例：计算 $g[i][3]$   
第*i*层共8本书

1	5	2	3	4	5	2	3
---	---	---	---	---	---	---	---

1	5	2	3	4	5	2	3
---	---	---	---	---	---	---	---

1	5	2	3	4	5	2	3
---	---	---	---	---	---	---	---

1	5	2	3	4	5	2	3
---	---	---	---	---	---	---	---

枚举所有连续和,求最小值

前缀和做差


预处理  
每一层

$g[i][w]$ 表示第 $i$ 层拿最多 $w$ 本时  
该层最多拿多少价值

```
11 void group(int i){  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21 }
```

预处理第 $i$ 层: $g[i][]$

$g[i][w]$ 表示第 $i$ 层拿最多 $w$ 本时  
该层最多拿多少价值

```
29 for(int i=1;i<=n;++i) group(i);
30 for(int i=1;i<=n;++i)
31     for(int j=m;j>=0;--j){
32         int T=min(c[i],j);
33         
34     }
35
36 cout<<f[m]<<endl;
```

## 建模练习

快快编程  
kkcoding.net



# 建模思维

快快909



分组01背包标准模块

请同学写出  
题目大意

# 分组01背包+909

多支股票

总投资次数不超过C次

不超过C段

求利润最高

$v[i][p]$ 代表第 $i$ 支股票分成不超过 $p$ 段时的最大盈利

分组收益函数 $v_i(p)$

泛化物品=1个分组

$g[i][p]$ 代表1个分组内物品

$(item)\{p, v[i][p]\}$

# 建模思维

快快1048



分组01背包标准模块

请同学写出  
题目大意

# 分组1背包+1048

多张地图/多关游戏

总撞墙次数不超过C次

到达所有终点求最少总步数

$v[i][p]$ 代表第 $i$ 张地图用不超过 $p$ 次撞墙到终点最少步数

分组收益函数 $v_i(p)$

泛化物品=1个分组

$g[i][p]$ 代表1个分组内物品

$(item)\{p, v[i][p]\}$

# 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

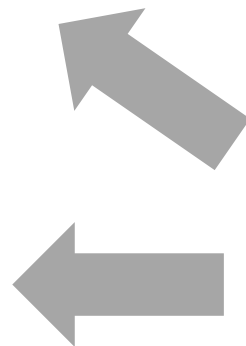
查错方法：

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

1936

1937

1938

拓展题

1166

自编1道"分组背包+"问题