

信奥算法

动态规划

Dynamic Programming

背包问题

快快编程
kkcoding.net

DP实战步骤

1

定义状态

2

优化状态

3

手动
填写
数组
所有
格子

4

再总结状态转移方程

5

再总结边界状态

6

优化
决策

kkcoding.net

复习01背包

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  const int MAXC=2009;
4  int n,c,w,v,f[MAXC];
5  int main(){
6      cin>>c>>n;
7      for(int i=1;i<=n;i++) {
8          cin>>w>>v;
9          for( )
10             f[j]=max(f[j], );
11     }
12     cout<<f[c]<<endl;
13     return 0;
14 }
```

从大到小
枚举载重j

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  const int MAXC=2009;
4  int n,c,w,v,f[MAXC];
5  void bb01(int w,int v){
6      for(int j=c;j>=w;j--){
7          f[j]=max(f[j],f[j-w]+v);
8      }
9  int main(){
10     cin>>c>>n;
11     for(int i=1;i<=n;i++){
12         cin>>w>>v;
13         bb01(w,v);
14     }
15     cout<<f[c]<<endl;
16     return 0;
17 }
```

分组01背包

一个背包最大载重 C ，现在有 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**。求解将哪些物品装入背包可使这些物品的总重量不超过背包载重且总价值最大。

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

3 3 1

4 8 2

6 9 2

2 8 3

3 9 3

输出样例

20

载重10,共6件物品,分组号{0,1,2,3}共4组

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

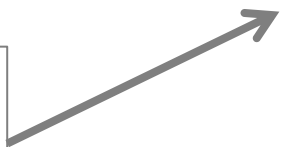
$$f[i][j] = \max\{$$

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

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

从大到小
枚举载重j

```
17 for(ll i=0;i<=g;i++)
18     for(ll j=c;j>=0;j--)
19         for(ll 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;
```

错误代码

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

错在哪?

下
越
界

数组编号
可能是负数

多重背包

N件物品放入载重为C的背包。每件物品最多有 m_i 件，第i件物品的重量为 w_i ，价值为 v_i 。选择将哪些物品放入背包令总重量不超过背包的载重且物品的价值总和最大。

第一行2个整数，表示物品种类数量n和背包容量C

下面n行，每行3个整数，分别表示第i种物品的重量 w_i ，价值 v_i 和数量 m_i

$n \leq 500, C \leq 10000, w_i \leq 100, v_i \leq 1000, m_i \leq 10$

输入样例

5 100

8 2 4

4 5 9

3 5 7

4 3 6

2 2 1

输出样例

104

共5件物品,载重100

重量8,价值2

重量4,价值5

重量3,价值5

重量4,价值3

重量2,价值2

多重背包

N 件物品放入载重为 C 的背包。每件物品最多有 m_i 件，第 i 件物品的费用（重量、体积等）为 w_i ，价值为 v_i 。选择将哪些物品放入背包令总费用不超过背包的载重且物品的价值总和最大。

简化版
找灵感

灵感
解锁

二进制
拆分

01背包

思考如何简化？

尝试012背包,给出算法

尝试0123背包,给出算法

同种物品
1件单品

同种物品
2件套装

3件套装
可以拼装

多重背包

N 件物品放入载重为 C 的背包。每件物品最多有 m_i 件，第 i 件物品的费用（重量、体积等）为 w_i ，价值为 v_i 。选择将哪些物品放入背包令总费用不超过背包的载重且物品的价值总和最大。

某种物品
有7件

1件
套装

2件
套装

4件
套装

某种物品
有10件

1件
套装

2件
套装

4件
套装

3件
套装

二进制
拆分

01背包

某种物品
有14件

1件
套装

2件
套装

4件
套装

7件
套装

多重背包 - 二进制拆分

```
5 void bb01(int w,int v){
6     for(int j=c;j>=w;j--){
7         f[j]=max(f[j],f[j-w]+v);
8     }
9 int main(){
10     cin>>n>>c;
11     for(int i=1;i<=n;i++){
12         cin>>w>>v>>s;
13         for(int k=1;k<=s;k*=2) bb01(k*w,k*v);
14         if(s) bb01(s*w,s*v);
15     }
16     cout<<f[c]<<endl;
17     return 0;
18 }
```

容易出题

部分分等级多

实用场景多

混合背包

N 件物品放入载重为 C 的背包。
第一类物品每种有无数件，共 n 种。
第二类物品每种有1件，共 m 种。
第三类物品每种有 a_i 件，共 k 种。

完全背包

从小到大
枚举载重 j

01背包

多重背包



01背包

二进制
拆分

从大到小
枚举载重 j

多重背包有可能变成完全背包吗?

足够多


快快编程1167

快快编程
kkcoding.net

```
7 void bb01(ll b, ll v){  
8     for(ll j=V; j>=b; --j)  
9         f[j]=max(f[j], f[j-b]+v);  
10 }
```

```
11 void bbINF(ll b, ll v){  
12     for(ll j=b; j<=V; ++j)  
13         f[j]=max(f[j], f[j-b]+v);  
14 }
```

```
18 cin>>n>>m>>k>>V;
19 for(ll i=1;i<=n+m+k;i++)cin>>b[i];
20 for(ll i=1;i<=n+m+k;i++)cin>>v[i];

21 for(ll i=1;i<=n;i++)bbINF(b[i],v[i]);
22 for(ll i=n+1;i<=n+m;i++)bb01(b[i],v[i]);
23 for(ll i=n+m+1;i<=n+m+k;i++){
24     cin>>a;
25     for(ll k=1;k<=a;a-=k,k*=2)bb01(k*b[i],k*v[i]);
26     
27 }

28 cout<<f[V]<<endl;
```

讨论题：凑硬币

硬币有 N 种不同的面额 $a[i]$ ，并且每种面额的硬币有一定的数量 $c[i]$ 。选取任意硬币，最多能组成几种总价值，且总价值不超过 m 。

多个测试点。每个测试点，第一行包含两个整数 $N(1 \leq N \leq 100)$ ， $M(M \leq 100000)$ ，第二行前 N 个数表示 N 种硬币的面额 $A[i](1 \leq A[i] \leq 100000)$ ，后 N 个数表示 N 种硬币的数量 $C[i](1 \leq C[i] \leq 100000)$ 。以0 0 结尾。

输入样例

3 10

1 2 4 2 1 1

2 5

1 4 2 1

0 0

输出样例

8

4



凑硬币

硬币有 N 种不同的面额 $a[i]$ ，并且每种面额的硬币有一定的数量 $c[i]$ 。选取任意硬币，最多能组成几种总价值，且总价值不超过 m 。

$h[i][j]$: 只用前 i 种硬币，总额恰好 j ，能否凑成

可行性问题

多重
凑数



二进制
拆分

01
凑数

复杂度?

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

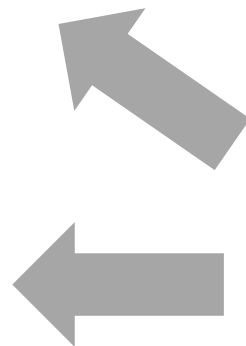
查错方法：

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



快快编程作业

135

1167

140

拓展题

1905, 1906, 1075