

太戈编程
etiger.vip

信奥算法

1870. 投资人



01背包变种

经典
方法

$f[i][j]$ 表示只考虑前 i 种物品
用载重 j 的包最多能装多少价值

难点

$W \leq 1000000000$
 W 特别大, $f[][]$ 会ME超出内存限制

方法1

控制 $f[][]$ 大小不ME, 只拿部分分

方法2

寻找突破口, 发现小数据:
 $n \leq 100$,
 $v[i] \leq 1000$

01背包变种

寻找突破口,发现小数据: $n \leq 100$, $v[i] \leq 1000$



状态定义
方法1

$g[i][p]$ 表示只考虑前 i 种物品
能装下至少价值为 p 的包最小载重是多少

状态定义
方法2

$h[i][p]$ 表示只考虑前 i 个物品
能装下恰好价值为 p 的包最小载重是多少

状态定义
方法1

$g[i][p]$ 表示只考虑前 i 种物品
能装下至少价值为 p 的包最小载重是多少

状态定义
方法2

$h[i][p]$ 表示只考虑前 i 个物品
能装下恰好价值为 p 的包最小载重是多少

```

1  /*姓名XXX
2  g[i][p]表示只考虑前i种物品
3  能装下价值至少为p的包最小载重是多少
4  初始化注意g[i][p]=INF 不是0
5  n=3,W=100
6  w[1]=50,v[1]=2
7  w[2]=60,v[2]=2
8  w[3]=40,v[3]=1
9      p=0,p=1,p=2,p=3,p=4,p=5
10 i=0   0   INF  INF  INF  INF  INF
11 i=1   0   50   50   INF  INF  INF
12 i=2   0   50   50   110  110  INF
13 i=3   0   40   50   ?    ?    ?
14 */

```

请完成1-14行注释

3分钟后老师检查

$g[i][p]$ 表示只考虑前 i 种物品
能装下至少价值为 p 的包最小载重是多少

状态定义
方法1

```
25 cin>>n>>W;  
26 for(ll i=1;i<=n;++i)cin>>w[i]>>v[i];  
27 ll V=n*1000;  
28 for(ll p=1;p<=V;++p)g[0][p]=INF; ←  
29 g[0][0]=0;  
30 for(ll i=1;i<=n;++i)  
31     for(ll p=0;p<=V;++p)  
32         if(v[i]>p)g[i][p]=  
33         else g[i][p]=min(g[i-1][p],
```

$g[i][p]$ 表示只考虑前 i 种物品
能装下至少价值为 p 的包最小载重是多少

状态定义
方法1

请同学思考如何输出答案

```
34 for(int p=V;p>=0;--p)
35     if() {
36         cout<<p<<endl;
37         break;
38     }
```


高频错误

数组大小定义错误, INF常量错误

```
17 typedef long long ll;  
18 const ll N=   
19 const ll MAXV=   
20 const ll INF=   
21 ll n, W, w[N], v[N], g[N][MAXV];
```

请同学写出
各个常量大小

对于100%数据, 保证 $n \leq 100$, $W \leq 1000000000$,
 $w[i] \leq 1000000000$, $v[i] \leq 1000$

状态定义
方法1

$g[i][p]$ 表示只考虑前 i 种物品
能装下至少价值为 p 的包最小载重是多少

状态定义
方法2

$h[i][p]$ 表示只考虑前 i 个物品
能装下恰好价值为 p 的包最小载重是多少

```

1  /*姓名XXX
2  h[i][p]表示只考虑前i个物品
3  能装下恰好价值为p的包最小载重是多少
4  初始化注意h[i][p]=INF 不是0
5  n=3, W=100
6  w[1]=50, v[1]=2
7  w[2]=60, v[2]=2
8  w[3]=40, v[3]=1
9      p=0, p=1, p=2, p=3, p=4, p=5
10 i=0    0    INF    INF    INF    INF    INF
11 i=1    0    INF    50    INF    INF    INF
12 i=2    0    INF    50    ?    ?    ?
13 i=3    0    40    50    90    110    150
14 */

```

请同学完成14行注释

3分钟后老师检查

$h[i][p]$ 表示只考虑前 i 个物品
能装下恰好价值为 p 的包最小载重是多少

状态定义
方法2

```
25 cin>>n>>W;  
26 for(ll i=1;i<=n;++i)cin>>w[i]>>v[i];  
27 ll V=n*1000;  
28 for(ll p=1;p<=V;++p)h[0][p]=INF; ←  
29 h[0][0]=0;  
30 for(ll i=1;i<=n;++i)  
31     for(ll p=0;p<=V;++p)  
32         if(v[i]>p)h[i][p]=  
33         else h[i][p]=min(h[i-1][p],
```

查错方法

```
39 // 查错打印:  
40 // for(LL i=0;i<=n;++i,cout<<endl)  
41 //     for(LL p=0;p<=V;++p)cout<<h[i][p]<<" ";
```

打印表格和手算表格
核对查错

理解背包模型

资源分配优化问题

约束条件

目标函数

决策

有某种稀缺资源待分配

最大化分配后的收益

选哪些对象组合

识别成本

识别收益

选择 i 号对象，代价 $w[i]$

选择 i 号对象，收获 $v[i]$

讨论

对于01背包模型
如何设计暴力枚举算法

用途1	用于检验DP的正确性
用途2	取得部分得分



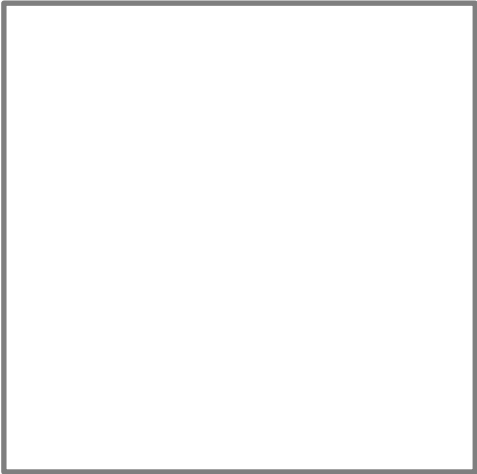
2794

暴力枚举

01决策：枚举每个人是否选取

复杂度 $O(2^n)$

讨论：如何剪枝？

```
10 int ans;  
11 int sZ,sR;  
12 void dfs(int id){  
13     if(id>n){  
14         if(sZ<0 || sR<0) return ;  
15           
16           
17     }  
18     dfs(id+1);  
19       
20  
21  
22  
23  
24 }
```

动态规划

请同学设计状态

A

$f[i][j][k]$ 表示只考虑前 i 个人时
能否凑出智商和恰为 j 且人品和恰为 k

B

$f[i][j]$ 表示只考虑前 i 个人时
智商和恰为 j 时人品和的最大值

C

$f[i][j]$ 表示只考虑前 i 个人时
智商和大于等于 j 时人品和的最大值

约束
目标
转换

原题的约束：人品和大于等于0
改成目标：最大化人品和

```

30  /*
31  f[i][j]表示只考虑前i个人时
32  智商和大于等于j时人品和的最大值
33  n=3
34  -5 1
35  2 -2
36  3 2
37  j=-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5
38  i=0  0  0  0  0  0  0  -  -  -  -  -
39  i=1  1  0  0  0  0  0  -  -  -  -  -
40  i=2  1  0  0  0  0  0  -2 -2  -  -  -
41  i=3
42  */

```

分析表格填写的要点

难点

数组编号会用到负数

map映射 代替 数组

```
45 map<int,int> f[N];
```

编号整体平移到非负数

```
76 int g[N][N*M*2];  
77 int id(int i){return i+N*M;};
```

难点

列数编号的预计算

47

48

49

50

51

```
int mx=0,mn=0;
for(int i=1;i<=n;i++){
    if(z[i]>0)mx+=z[i];
    else 
}
```

```

52 for(int j=1;j<=mx;j++) f[0][j]=
53 for(int i=1;i<=n;i++)
54     for(int j=mn;j<=mx;j++){
55         f[i][j]=f[i-1][j];
56         int pre=j-z[i];
57         if(pre>mx)continue;
58         if(pre<mn)
59         if(f[i-1][pre]==-INF)
60         f[i][j]=max(f[i][j],
61     }

```

第58行要注意
 无论怎么选
 智商和 $\geq mn$
 一定能够满足

map映射 代替 数组

时间复杂度 $O(n*n*R*\log(n*R))$

$n=100, R=1000*2$

还是会超时

编号整体平移到非负数

```
76 int g[N][N*M*2];  
77 int id(int i){return i+N*M;};
```

所有 $g[][j]$
都改成 $g[][id(j)]$
包括预处理/转移方程/输出答案

```
84 for(int j=1;j<=mx;j++) g[0][id(j)]=-INF;
85 for(int i=1;i<=n;i++)
86     for(int j=mn;j<=mx;j++){
87         g[i][id(j)]=g[i-1][id(j)];
88         int pre=j-z[i];
89         if(pre>mx)continue;
90         if(pre<mn)pre=mn;
91         if(g[i-1][id(pre)]==-INF)continue;
92         g[i][id(j)]=max(g[i][id(j)],g[i-1][id(pre)]+r[i]);
93     }
```

```
94 ans=0;
95 for(int j=0;j<=mx;j++){
96     
97     ans=max(ans, 
98 }
99 cout<<ans<<endl;
```

时间复杂度?

$O(n*n*R)$

$n=100, R=1000*2$

大小数据分离 确保部分分

```
110 input();  
111 if(n<=20)  
112     solveBF();  
113 else if(n<=30)  
114     solveMap();  
115 else  
116     solve();
```

太戈编程

1870,2794