

完全背包问题

有 N 种物品和一个容量是 V 的背包，每种物品都有无限件可用。

第 i 种物品的体积是 v_i ，价值是 w_i 。

求解将哪些物品装入背包，可使这些物品的总体积不超过背包容量，且总价值最大。
输出最大价值。

输入格式

第一行两个整数， N ， V ，用空格隔开，分别表示物品种数和背包容积。

接下来有 N 行，每行两个整数 v_i, w_i ，用空格隔开，分别表示第 i 种物品的体积和价值。

输出格式

输出一个整数，表示最大价值。

数据范围

$0 < N, V \leq 1000$

$0 < v_i, w_i \leq 1000$

输入样例

```
4 5
1 2
2 4
3 4
4 5
```

输出样例：

```
10
```

理解

集合: 所有只从前*i*个物品中选, 总性能不超过*j*的方案集合

状态表示: $f(i, j)$

属性: Mon

状态计算

$v_i \leq j \Rightarrow v_{i-1} \leq j$

$f(i-1, j)$

$f(i, j) = \max(f(i-1, j), f(i, j-v_i) + w_i)$

$f(i, j) = \max(f(i-1, j), f(i-1, j-v_i) + w_i, f(i-1, j-2v_i) + 2w_i, \dots)$

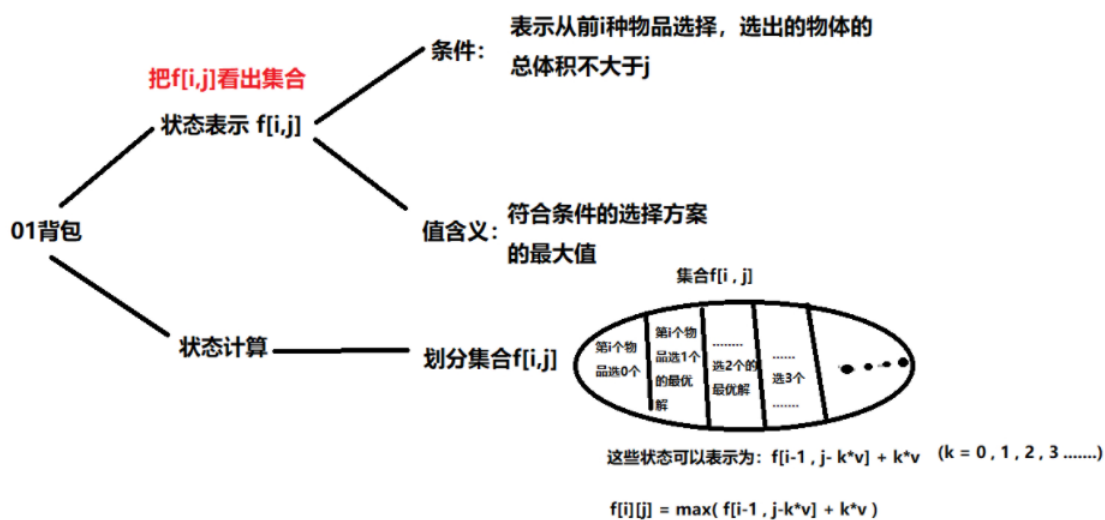
$f(i, j-v) = \max(f(i-1, j-v), f(i-1, j-2v) + w, f(i-1, j-3v) + 2w, \dots) + w$

QQ交流
WW

```

1. /*
2 1. 01背包: f[i][j] = max(f[i-1][j], f[i-1][j-v] + w);
3 2. 完全背包: f[i][j] = max(f[i-1][j], f[i][j-v] + w);
4 */
    
```

背包问题



代码

```

#include<iostream>
using namespace std;
const int N = 1010;
int f[N][N];
int v[N], w[N];
int main()
{
    int n, m;
    
```

```

cin>>n>>m;
for(int i = 1 ; i <= n ; i ++ )
{
    cin>>v[i]>>w[i];
}

for(int i = 1 ; i<=n ; i++)
for(int j = 0 ; j<=m ; j++)
{
    for(int k = 0 ; k*v[i]<=j ; k++)
        f[i][j] = max(f[i][j],f[i-1][j-k*v[i]]+k*w[i]);
}

cout<<f[n][m]<<endl;
}

```

作者: Aniway

链接: <https://www.acwing.com/solution/content/5345/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

```

9
10 const int N = 1010;
11
12 int n, m;
13 int v[N], w[N];
14 int f[N][N];
15
16 int main()
17 {
18     cin >> n >> m;
19
20     for (int i = 1; i <= n; i ++ ) cin >> v[i] >> w[i];
21
22     for (int i = 1; i <= n; i ++ )
23         for (int j = 1; j <= m; j ++ )
24         {
25             f[i][j] = f[i - 1][j];
26             if (j >= v[i]) f[i][j] = max(f[i][j], f[i][j - v[i]] + w[i]);
27         }
28
29     cout << f[n][m] << endl;
30
31     return 0;

```