

01背包问题

有 N 件物品和一个容量是 V 的背包。每件物品只能使用一次。

第 i 件物品的体积是 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，分别表示物品数量和背包容积。

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 件物品的体积和价值。

输出格式

输出一个整数，表示最大价值。

数据范围

$0 < N, V \leq 1000$

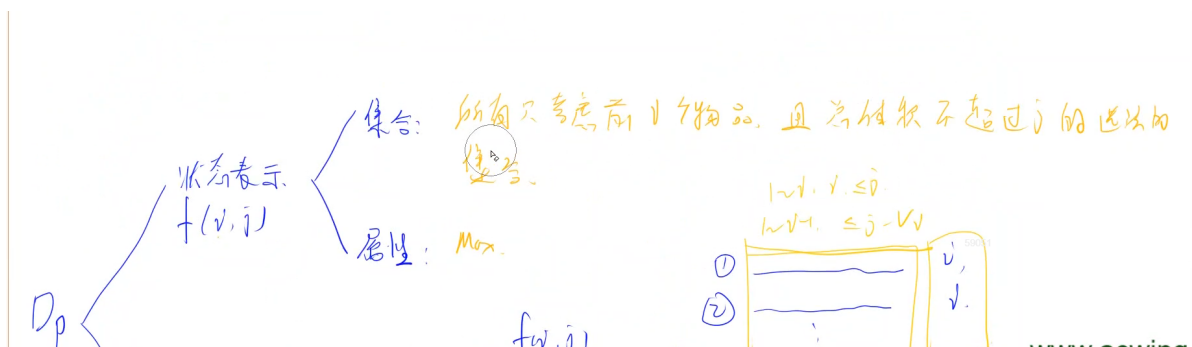
$0 < v_i, w_i \leq 1000$

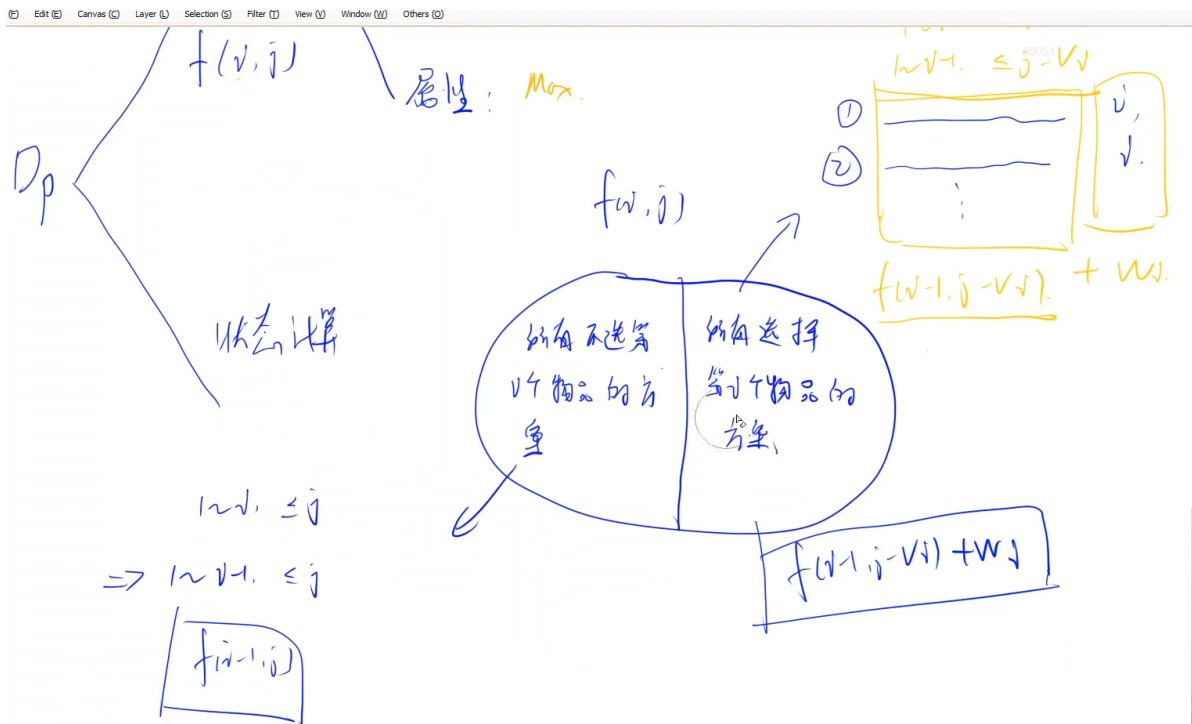
输入样例

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

输出样例：

```
8
```





代码:

```
//无优化版
#include <iostream>

using namespace std;

const int N = 1010;

int n, m;
int v[N], w[N];
int f[N][N];

int main() {
    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++) {
            f[i][j] = f[i-1][j];
            if(j >= v[i]) f[i][j] = max(f[i][j], f[i-1][j-v[i]]+w[i]);
        }
    }

    cout << f[n][m] << endl;
    return 0;
}
```

//有优化版

/*

1. $f[i]$ 仅用到了 $f[i-1]$ 层,
2. j 与 $j-v[i]$ 均小于 j
3. 若用到上一层的状态时, 从大到小枚举, 反之从小到大哦

*/

```
#include <iostream>
```

```
using namespace std;

const int N = 1010;

int n, m;
int v[N], w[N];
int f[N];

int main() {
    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 = m; j >= v[i]; j--)
            f[j] = max(f[j], f[j-v[i]]+w[i]);
    cout << f[m] << endl;
    return 0;
}
```

作者：竹林正在青

链接：<https://www.acwing.com/solution/content/4515/>

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。