

## 分组背包问题：

有  $N$  件物品和一个容量为  $V$  的背包。第  $i$  件物品的费用是  $C_i$ ，价值是  $W_i$ 。这些物品被划分为  $K$  组，每组中的物品互相冲突，最多选一件。求解所有合法方案的最大价值和。

### 分析

- 首先背包问题的最小规模解都是知道的。
- 状态转移方程
  - 定义  $f_{i,j}$  指标函数为，用容量为  $j$  的背包，选择前  $i$  组物品。所有合法方案的最大价值和问题。
$$f_{i,j} = \max(f_{i-1,j}, f_{i-1,j-w_{s_k}} + v_k), 0 \leq k \leq s, j - W_k \geq 0$$
- 最终状态的复杂度为  $O(m \times \sum s_i)$

### code

```
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;
struct node
{
    int w;
    int v;
};

const int maxn = 110;
int f[maxn];

int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
    {
        int s;
        cin >> s;
        vector<node> a;
        int minw = 0;
        for (int i = 0; i < s; i++)
        {
            int w, v;
            cin >> w >> v;
            minw = min(w, minw);
            a.push_back({w, v});
        }
        for (int j = m; j >= minw; j--)
```

```

        for (int k = 0; k < a.size(); k++)
            if (j >= a[k].w)
                f[j] = max(f[j], f[j - a[k].w] + a[k].v);
    }
}
cout << f[m] << '\n';
}

```

## 改进

- 由于std常数比较大，必要时开数组来记录同一组的物品的属性。

```

#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

const int maxn = 110;
int v[maxn];
int w[maxn];
int f[maxn];
int main()
{
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
    {
        int s;
        cin >> s;
        int minw = 0;
        for (int i = 0; i < s; i++)
        {
            cin >> w[i] >> v[i];
            minw = min(w[i], minw);
        }
        for (int j = m; j >= minw; j--)
        {
            for (int k = 0; k < s; k++)
                if (j >= w[k])
                    f[j] = max(f[j], f[j - w[k]] + v[k]);
        }
    }
    cout << f[m] << '\n';
}

```