

二维费用背包问题

问题

二维费用的背包问题是指：对于每件物品，具有两种不同的费用；选择这件物品必须同时付出这两种代价；对于每种代价都有一个可付出的最大值（背包容量）。问怎样选择物品可以得到最大的价值。设这两种代价分别为代价1和代价2，第*i*件物品所需的两种代价分别为 $w[i]$ 和 $g[i]$ 。两种代价可付出的最大值（两种背包容量）分别为*V*和*T*。物品的价值为 $v[i]$ 。

算法

费用加了一维，只需状态也加一维即可。设 $f[i][j][k]$ 表示前*i*件物品付出两种代价分别为*j*和*k*时可获得的最大价值。状态转移方程就是：

$$f[i][j][k] = \max(f[i - 1][j][k], f[i - 1][j - w[i]][k - g[i]] + v[i])$$

如前述方法，可以只使用二维的数组：当每件物品只可以取一次时变量*j*和*k*采用逆序的循环，当物品有如完全背包问题时采用顺序的循环。当物品有如多重背包问题时拆分物品。代码：

题目描述

有*N*件物品和一个容量是*V*的背包，背包能承受的最大重量是*M*。

每件物品只能用一次。体积是 v_i ，重量是 m_i ，价值是 w_i 。

求解将哪些物品装入背包，可使物品总体积不超过背包容量，总重量不超过背包可承受的最大重量，且价值总和最大。 输出最大价值。

输入格式

第一行三个整数，*N*,*V*,*M*，用空格隔开，分别表示物品件数、背包容积和背包可承受的最大重量。

接下来有*N*行，每行三个整数 v_i, m_i, w_i ，用空格隔开，分别表示第*i*件物品的体积、重量和价值。

输出格式

输出一个整数，表示最大价值。

数据范围

$$0 < N \leq 1000$$

$$0 < V, M \leq 100$$

$$0 < v_i, m_i \leq 100$$

$$0 < w_i \leq 1000$$

输入样例

```
4 5 6
1 2 3
2 4 4
3 4 5
4 5 6
```

输出样例：

```
8
```

In [1]:

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef __cplusplus //曾经的C/C++, 使用这个宏
#define Max(a, b) ((a > b) ? (a) : (b))
extern "C" {
    void _2pack() {
        freopen("dp04beibao05_01.in", "r", stdin);
        int ArrMax{110};
        int dp[ArrMax][ArrMax];
        int N, V, M;
        scanf("%d %d %d", &N, &V, &M);
        while (N--) {
            int v, m, w;
            scanf("%d %d %d", &v, &m, &w);
            for (int ind1 = V; ind1 >= v; --ind1)
                for (int ind2 = M; ind2 >= m; --ind2)
                    dp[ind1][ind2] = Max(dp[ind1][ind2], dp[ind1 - v][ind2 - m] + w);
        }
        printf("%d", dp[V][M]);
    }
}
#endif
```

Out[1]:

In [2]:

```
_2pack();
```

8

Out[2]:

(void) nullptr

物品总个数的限制

有时，“二维费用”的条件是以这样一种隐含的方式给出的：最多只能取 M 件物品。这事实上相当于每件物品多了一种“件数”的费用，每个物品的件数费用均为1，可以付出的最大件数费用为 M 。换句话说，设 $f[i][j]$ 表示付出费用 i 、最多选 j 件时可得到的最大价值，则根据物品的类型（01、完全、多重）用不同的方法循环更新，最后在 $f[0 \cdots V][0 \cdots M]$ 范围内寻找答案。

复数域上的背包问题

另一种看待二维背包问题的思路是：将它看待成复数域上的背包问题。也就是说，背包的容量以及每件物品的费用都是一个复数。而常见的一维背包问题则是实数域上的背包问题。（注意：上面的话其实不严谨，因为事实上我们处理的都只是整数而已。）所以说，一维背包的种种思想方法，往往可以应用于二位背包问题的求解中，因为只是数域扩大了而已。

例题：

[Luogu 1507 NASA的食物计划 \(https://www.luogu.org/problemnew/show/P1507\)](https://www.luogu.org/problemnew/show/P1507)
[HDU 2159 FATE \(http://acm.hdu.edu.cn/showproblem.php?pid=2159\)](http://acm.hdu.edu.cn/showproblem.php?pid=2159)

In []: