

## 计算机实验

解答. [0/1 背包问题] 设  $dp[i][j]$  表示从前  $i$  个物品中, 使用容量为  $j$  的背包所能获得的最大价值. 则状态转移方程如下

$$dp[i][j] = \max\{dp[i-1][j-w[i]] + v[i], dp[i-1][j]\}.$$

其中  $\max$  中的第一项  $dp[i-1][j-w[i]] + v[i]$  表示在剩余容量为  $j$  的前提下, 选择第  $i$  个物品所获得的最大价值, 则可以从前  $i-1$  个物品容量为  $j-w[i]$  所能获得的最大价值加上当前物品的价值  $v[i]$  得到;  $\max$  的第二项  $dp[i-1][j]$  表示不选取第  $i$  件物品, 则直接从前  $i-1$  件物品容量为  $j$  的最大价值直接转移得到.

若  $dp[i][j]$  是从  $\max$  的第一项转移得到, 则选择了当前物品  $i$ ; 否则, 从  $\max$  的第二项转移得到, 则没有选择物品  $i$ . 通过记录  $dp[i][j]$  从谁转移得到的, 即可得知最终选取的物品是哪些. 最终输出可逆向枚举得到.

实现代码如下:

---

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 const int N = 1001;
4 int w[N], v[N];
5 int dp[N][N];
6 bool use[N][N];
7 int main() {
8     // freopen("tmp.in", "r", stdin);
9     int tot, n;
10    cin >> tot >> n;
11    for (int i = 1; i <= n; i++) cin >> w[i];
12    for (int i = 1; i <= n; i++) cin >> v[i];
13    for (int i = 1; i <= n; i++) {
14        for (int j = w[i]; j <= tot; j++) {
15            dp[i][j] = dp[i-1][j];
16            if (dp[i-1][j-w[i]] + v[i] > dp[i][j]) {
17                dp[i][j] = dp[i-1][j-w[i]] + v[i];
18                use[i][j] = 1; // 获得当前物品
19            }
20        }
21    }
```

```

22     cout << " 能获得的最大价值为: " << dp[n][tot] << '\n';
23     int now = n, less = tot;
24     cout << " 选取物品编号: ";
25     while (now) {
26         if (use[now][less]) {
27             cout << now << ' ';
28             less -= w[now];
29         }
30         now--;
31     }
32     cout << '\n';
33     return 0;
34 }
35
36 #if 0
37 输入样例:
38 10
39 5
40 2 2 6 5 4
41 6 3 5 4 6
42 输出:
43 能获得的最大价值为: 15
44 选取物品编号: 5 2 1
45 #endif

```

---