

动态规划 2

二进制优化

算法模型

给定一个容量为 m 的背包和 n 种物品。每种物品价值为 v_i 重量为 w_i 数量为 c_i 求最多可以得到的价值。

算法实现

暴力方法为将每种物品转化为 c_i 个独立物品考虑，然后就是一个 01 背包问题。

不难发现任意一个不超过 c_i 的正整数一定用 $1, 2, 4, \dots, 2^{n-1}, (c_i - 2^{n-1} + 1)$ 表示。

于是可以将每种物品拆成 $1, 2, 4, \dots, 2^{n-1}, (c_i - 2^{n-1} + 1)$ 个物品构成的包考虑。

物品总数优化为 $O(\sum_{i=1}^n \log \{c_i\})$ 总时间复杂度为 $O(m \sum_{i=1}^n \log \{c_i\})$

代码模板

```
const int MAXM=4e4+5,MAXC=2005;
int dp[MAXM],v[MAXC],w[MAXC];
int main()
{
    int n=read_int(),m=read_int(),cnt=0;
    _for(i,0,n){
        int a=read_int(),b=read_int(),c=read_int();
        for(int j=1;j<c;j<=1){
            v[cnt]=j*a,w[cnt++]=j*b;
            c-=j;
        }
        v[cnt]=c*a,w[cnt++]=c*b;
    }
    _for(i,0,cnt)
        for(int j=m;j>=w[i];j--)
            dp[j]=max(dp[j],dp[j-w[i]]+v[i]);
    enter(dp[m]);
    return 0;
}
```

Last
update: 2020-2021:teams:legal_string:jxm2001: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92_2&rev=1603175640
2020/10/20 动态规划_2
14:34

From:
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92_2&rev=1603175640 

Last update: **2020/10/20 14:34**