混合背包问题

问题

如果将前面三个背包混合起来，也就是说，有的物品只可以取一次（01背包），有的物品可以取无限次（完全背包），有的物品可以取的次数有一个上限（多重背包），应该怎么求解呢?

01背包与完全背包的混合

考虑到在01背包和完全背包中给出的伪代码只有一处不同，故如果只有两类物品：一类物品只能取一次，另一类物品可以取无限次，那么只需在对每个物品应用转移方程时，根据物品的类别选用顺序或逆序的循环即可，复杂度是$O(VN)$。

再加上多重背包

如果再加上有的物品最多可以取有限次，那么原则上也可以给出$O(VN)$的解法：遇到多重背包类型的物品用单调队列(**这里就先不讨论单调队列了**)解即可。但如果不考虑超过$NOIP$范围的算法的话，用多重背包中将每个这类物品分成$O(\log(p[i]))$个01背包的物品的方法也已经很优了。当然，更清晰的写法是调用我们前面给出的三个相关过程。代码：

题目描述

有N种物品和一个容量是V的背包。

物品一共有三类:

第一类物品只能用1次（01背包）；

第二类物品可以用无限次（完全背包）；

第三类物品最多只能用s_i次（多重背包）；

每种体积是v_i，价值是w_i。

求解将哪些物品装入背包，可使物品体积总和不超过背包容量，且价值总和最大。

输出最大价值。

输入格式

第一行两个整数，N，V，用空格隔开，分别表示物品种数和背包容积。

接下来有N行，每行三个整数v_i,w_i,s_i，用空格隔开，分别表示第i种物品的体积、价值和数量。

$s_i=-1$表示第i种物品只能用1次；

$s_i=0$表示第i种物品可以用无限次；

$s_i>0$表示第i种物品可以使用s_i次；

输出格式

输出一个整数，表示最大价值。

数据范围

$0<N,V\leq 1000$

$0<v_i,w_i\leq 1000$

$-1\leq s_i\leq 1000$

输入样例

```
4 5
1 2 -1
2 4 1
3 4 0
4 5 2
```

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
extern "C" {
    using namespace std;
    const int maxn = (int)1e2+5;

    int val[maxn],cst[maxn],sze[maxn],N,V;
    int dp[maxn];

    int max(int a, int b){
        return a>=b?a:b;
    }
    void ZeroOnePack (int cost, int value) {   //01背包   逆序
        for (int i = V; i >= cost; i--) {
            dp[i] = max(dp[i], dp[i-cost]+value);
        }
    }

    void CompletePack (int cost, int value) {  //完全背包  顺序
        for (int i = cost; i <= V; i++) {
            dp[i] = max(dp[i], dp[i-cost]+value);
        }
    }

    void MultiplePack (int idx) {
        if(sze[idx]==-1){
            ZeroOnePack(cst[idx], val[idx]);
            return ;
        }
        if (sze[idx]*cst[idx] >= V || sze[idx]==0) {     //如果装不下，那就是完全背包问题
            CompletePack(cst[idx], val[idx]);  //调用完全背包
            return ;
        }
        int x = 1;
        int num = sze[idx];
        while (x <= num) {
            ZeroOnePack(x*cst[idx], x*val[idx]);
            num -= x;     //num-  1,2,4,8,16.....
            x <<= 1;    //x=1,2,4,8,16.....
        }
        if (num > 0) {  //如果还有剩则在单独考虑一次。
            ZeroOnePack(num*cst[idx], num*val[idx]);
        }
    }
    void print_result(){
        freopen("dp04beibao04_01.in","r",stdin);
        scanf("%d %d",&N,&V);
        for (int i = 0; i < N; i++) {
            scanf("%d %d %d",&cst[i],&val[i],&sze[i]);  //费用，价值，数量
        }
        memset(dp, 0, sizeof(dp));
        for (int i = 0; i < N; i++) {
            MultiplePack(i);           //多重背包
        }
        printf("%d",dp[V]);
    }
}
#endif
```

Out[1]:

In [3]:
```
print_result();
```

```
8
```

Out[3]: (void) nullptr

例题：

Luogu P1833 樱花 (https://www.luogu.org/problemnew/show/P1833)
HDU 3535 AreYouBusy (http://acm.hdu.edu.cn/showproblem.php?pid=3535)

In [ ]: