

# Multi University 2012 #5 B "Gold Miner" Solution

本作品採用[知識共享署名-非商业性使用-相同方式共享 3.0 Unported](#) 許可協議进行許可

write by Gestalti Lur

2012-09-11

題目鏈接：<http://acm.hdu.edu.cn/showproblem.php?pid=4341>

## 題目大意

直角座標系內有  $N(N \leq 200)$  個點  $P(x,y)$  ( $|x| \leq 200, -200 \leq y \leq 200$ )，從原點出發取點  $P_i$  需要  $t_i$  的時間，該點的價值為  $v_i$ ，如果從原點出發與某個點的連線和原點到其他點共線，則必須先取原點距離較短的點。

## 算法分析

可以看出來是一個很明顯的背包模型。共線的情況可以預處理一下合併為一個物品。

首先按照到原點的距離給所有點排序，然後處理共線的情況。設  $v[k, j]$  和  $w[k, j]$  分別表示取第  $k$  組共線的物品的第  $j$  個的時間和能夠取到的價值， $f[i]$  表示時間為  $i$  時取到的最大價值，則有：

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

關於這類複合物品的揹包問題 DD\_engi 在揹包問題九講<sup>1</sup>的泛化物品一節有詳細敘述。

## 參考代碼

```
/*
Multi-university 2012 #5 B
2012-09-11
ACCEPTED
gestapolur
*/
#include<cstdio>
#include<cstring>
#define MAXN 202
#define MAXV 40002

int v[ MAXN ][ MAXN ], w[ MAXN ][ MAXN ]; //v - cost , w - value
int tx[ MAXN ], ty[ MAXN ], tv[ MAXN ], tw[ MAXN ], cnt[ MAXN ],
dist[ MAXN ];
int f[ MAXV ], ans;
bool bel[ MAXN ];
int n , m , tn , tot;

void dp()
{
    int i , j , k;
    for( i = 1 ; i <= tn ; ++ i )
        for( j = m ; j ; -- j )
```

<sup>1</sup><https://github.com/tianyicui/pack>

```

    {
        if( j - v[ i ][ 1 ] < 0 ) break;
        for( k = 0 ; k <= cnt[ i ] ; ++ k )
            if( j - v[ i ][ k ] < 0 ) break;
            else if( f[ j ] < f[ j - v[ i ][ k ] ] + w[ i ][ k ] )
                {
                    f[ j ] = f[ j - v[ i ][ k ] ] + w[ i ][ k ];
                    ans = ans > f[ j ] ? ans : f[ j ];
                }
    }
    return ;
}

void slope( int i )
{
    for( int j = i + 1 ; j <= n ; ++ j )//check co-line case
        if( tx[ i ] * ty[ j ] == tx[ j ] * ty[ i ] )
            {
                bel[ j ] = true;
                ++ cnt[ tn ];
                v[ tn ][ cnt[ tn ] ] = v[ tn ][ cnt[ tn ] - 1 ] + tv[ j ];
                w[ tn ][ cnt[ tn ] ] = w[ tn ][ cnt[ tn ] - 1 ] + tw[ j ];
            }
    return ;
}

void init()
{
    int i , j , t;
    tn = 0;
    ans = 0;
    for( i = 1 ; i <= n ; ++ i )
        {
            scanf( "%d %d %d %d" , &tx[ i ] , &ty[ i ] , &tv[ i ] , &tw[ i ] );
            dist[ i ] = tx[ i ] * tx[ i ] + ty[ i ] * ty[ i ];
            cnt[ i ] = 0;
            bel[ i ] = false;
        }
    memset( f , 0 , sizeof( int ) * ( m + 1 ) );

    //sort
    for( i = 2 ; i <= n ; ++ i )
        for( j = 1 ; j < i ; ++ j )
            if( dist[ j ] > dist[ i ] )
                {
                    t = tx[ i ]; tx[ i ] = tx[ j ]; tx[ j ] = t;
                    t = ty[ i ]; ty[ i ] = ty[ j ]; ty[ j ] = t;
                    t = tw[ i ]; tw[ i ] = tw[ j ]; tw[ j ] = t;
                }

```

```

        t = tv[ i ]; tv[ i ] = tv[ j ]; tv[ j ] = t;
        t = dist[ i ]; dist[ i ] = dist[ j ]; dist[ j ] = t;
    }
//merge
for( i = 1 ; i <= n ; ++ i )
    if( not bel[ i ] )
    {
        bel[ i ] = true;
        cnt[ ++ tn ] = 1;
        v[ tn ][ 1 ] = tv[ i ];
        w[ tn ][ 1 ] = tw[ i ];
        slope( i );
    }
return ;
}

int main()
{
    while( scanf( "%d %d" , &n , &m ) not_eq EOF )
    {
        init();
        dp();
        printf("Case %d: %d\n" , ++ tot , ans );
    }
    return 0;
}

```