原创 Proud Merchants HDU - 3466（与顺序有关的01背包）

2019-09-27 19:20:31 Laaahu_ 阅读数 8 更多

题目链接：vj里面的题目
题意:
n件商品每件商品有三个属性，p代表的价格，q代表你至少要有多少钱才可以买这个商品q>p, v代表的是这个商品的价值，总共有m的钱，问最多可以
值之和是多少。
思路:
假设有A B 两个商品要买pa=5,qa=6;pb=5,qb=9,m=10;
如果要从状态转移得到A是否购买，因为条件的限制关系所以 A 只有在 m=10,9,8,7,6时才会发生状态转移。
而对于B来说就是m=10，9时才会有状态转移。
第二个物品要不要买要借助于第一个物品的状态转移之后的结果，所以必须先买A再买B，如果先买B对于A来讲m=6,7,8时转台转移就没有和有所关联.
所以我们要选择合适的顺序去跑背包。
当然也与价格有关系，当限定相同价格的物品出售的顺序不同时结果也不同，
此题策略：**限制小的价格贵的先选**
也就是q-p小的先选，
A： pa,qa;
B : pb,qb;
先选A至少要pa+qb的容量，先选B至少要pb+qa的容量,
如果pa+qb>pb+qa,那么要两个都选你只能先选A再选B。例如上面的AB例子。所以 按照 qa-pa<qb-pb(q1-p1<q2-p2)排序之后跑背包即可。

```
/*两物品A(p1,q1) B(p2 q2) 先选A则至少需要p1+q2的容量 先选B则至少需要p2+q1 如果p1+q2>p2+q1
  那么要选两个的话的就要先选A再选B 公式可换成q1-p1<q2-p2*/
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <cmath>
using namespace std;
const int maxn = 510;
const int M = 5e3+10;
struct node{
    int p,q,w;
    friend bool operator < (node a,node b){
        return (a.q-a.p)<(b.q-b.p);
    }
}arr[maxn];
int dp[M];
int main()
{
    int n,m;
    while(~scanf("%d%d",&n,&m)){
        for(int i=1;i<=n;i++){
            scanf("%d%d%d",&arr[i].p,&arr[i].q,&arr[i].w);
        }
        sort(arr+1,arr+n+1);
        memset(dp,0,sizeof(dp));
        for(int i=1;i<=n;i++){
            for(int j=m;j>=arr[i].q;j--){
                dp[j]=max(dp[j],dp[j-arr[i].p]+arr[i].w);
            }
        }
        printf("%d\n",dp[m]);
    }
}
```