

1. 给定 $n$ 个物品，物品价值分别为 $P_1, P_2, \dots, P_n$ ，物品重量分别 $W_1, W_2, \dots, W_n$ ，背包容量为 $M$ ，每种物品可部分装入到背包中。输出 $X_1, X_2, \dots, X_n$ ，其中， $0 \leq X_1 \leq 1$ ，使得 $\sum_{1 \leq i \leq n} P_i X_i$ 最大，且 $\sum_{1 \leq i \leq n} W_i X_i \leq M$ ，试设计一个算法求解该问题，并分析算法的正确性。

# 价值数组P 重量数组W

```
items = [(P[i], W[i], i) for i in range(n)]  
# 按照单位价值对 items 进行排序  
items.sort(key=lambda x: x[0]/x[1], reverse=True)
```

# 物品选择数组X

```
remain = M    total = 0  
for p, w, i in items:  
    if remain > w:  
        remain = remain - w  
        X[i] = 1  
        total = total + p.  
    else:  
        X[i] = remain / w  
        total += X[i] * p.  
        break  
return X, total
```

正确性：每次都装最贵的，保证质量不超标的情况下，价值也是最大的。

6. 假设有数目 unlimited 的面值为 25 美分, 10 美分, 5 美分, 1 美分的硬币, 请使用最少个数的硬币凑出 3.33 美元

```
def coin-change-greedy(amount, coins):
```

```
    count = 0
```

```
    for coin in coins:
```

```
        count += amount // coin
```

```
        amount %= coin
```

```
    return count
```