BTH004-背包问题之动态规划

姓名：赵水 学号：201806150329

1 问题：

给定背包的容量和一些物品的价值与重量，求背包能够装下的最大价值总量

2 算法：

采用DP（动态规划）算法。对每个物品进行分析拿与不拿取其较大者作为当前dp值，依次迭代求出最大值。

3 代码：

```
def dpForKnapsacksProblem(capacity,s,v):
    s.insert(0,0)
    v.insert(0,0)
    dp = []
    for i in range(len(s)):
        temp = [0] * (capacity+1)
        dp.append(temp)

    for i in range(1,len(dp)):
        for j in range(1,len(dp[i])):
            if j < s[i]:
                dp[i][j] = dp[i-1][j]
            else:
                dp[i][j] = max(dp[i-1][j - s[i]]+v[i],dp[i-1][j])
    return max(dp[len(v)-1]),dp
```

4 测试

4.1 代码

```
capacity = 17
s = [7, 4, 3, 10, 6, 9, 9]
v = [7, 9, 9, 8, 2, 9, 8]
print(dpForKnapsacksProblem(capacity,s.copy(),v.copy()))
```

4.2 输出

```
(27, [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0,
0, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7], [0, 0, 0, 0, 9, 9, 9, 9, 9, 9, 9, 16, 16,
16, 16, 16, 16, 16], [0, 0, 0, 9, 9, 9, 9, 18, 18, 18, 18, 18, 18, 18, 25, 25,
25, 25], [0, 0, 0, 9, 9, 9, 9, 18, 18, 18, 18, 18, 18, 18, 25, 25, 25, 26], [0,
0, 0, 9, 9, 9, 9, 18, 18, 18, 18, 18, 18, 20, 25, 25, 25, 26], [0, 0, 0, 9, 9,
9, 9, 18, 18, 18, 18, 18, 18, 20, 25, 25, 27, 27], [0, 0, 0, 9, 9, 9, 9, 18, 18,
18, 18, 18, 18, 20, 25, 25, 27, 27]])
```