# 动态规划 (DP) 算法学习 - 简书

原文：
http://www.cnblogs.com/steven_oyj/archive/2010/05/22/1741374.html

## 要解决的问题

动态规划算法要解决的是多阶段决策问题

```
初始状态→|决策 1 |→|决策 2 |→...→|决策 n |→结束状态
```

## 适用范围

1. 最优化原理：如果问题的最优解所包含的子问题的解也是最优的，就称该问题具有最优子结构，即满足最优化原理。

2. 无后效性：即某阶段状态一旦确定，就不受这个状态以后决策的影响。也就是说，某状态以后的过程不会影响以前的状态，只与当前状态有关。

3. 有重叠子问题：即子问题之间是不独立的，一个子问题在下一阶段决策中可能被多次使用到。（该性质并不是动态规划适用的必要条件，但是如果没有这条性质，动态规划算法同其他算法相比就不具备优势）

## 基本思想

将待求解的问题分解为若干个子问题（阶段），按顺序求解子阶段，前一子问题的解，为后一子问题的求解提供了有用

的信息。在求解任一子问题时，列出各种可能的局部解，通过决策保留那些有可能达到最优的局部解，丢弃其他局部解。依次解决各子问题，最后一个子问题就是初始问题的解。

由于动态规划解决的问题多数有重叠子问题这个特点，为减少重复计算，对每一个子问题只解一次，将其不同阶段的不同状态保存在一个二维数组中。

与分治法最大的差别是：适合于用动态规划法求解的问题，经分解后得到的子问题往往不是互相独立的（即下一个子阶段的求解是建立在上一个子阶段的解的基础上，进行进一步的求解）。

## 求解步骤

1. 划分阶段：按照问题的时间或空间特征，把问题分为若干个阶段。在划分阶段时，注意划分后的阶段一定要是有序的或者是可排序的，否则问题就无法求解。

2. 确定状态和状态变量：将问题发展到各个阶段时所处于的各种客观情况用不同的状态表示出来。当然，状态的选择要满足无后效性。

3. 确定决策并写出状态转移方程：因为决策和状态转移有着天然的联系，状态转移就是根据上一阶段的状态和决策来导出本阶段的状态。所以如果确定了决策，状态转移方程也就可写出。但事实上常常是反过来做，根据相邻两个阶段的状态之间的关系来确定决策方法和状态转移方程。

4. 寻找边界条件：给出的状态转移方程是一个递推式，需要一个递推的终止条件或边界条件。

一般，只要解决问题的阶段、状态和状态转移决策确定了，就可以写出状态转移方程（包括边界条件）。

实际应用中可以按以下几个简化的步骤进行设计：

1. 分析最优解的性质，并刻画其结构特征。
2. 递归的定义最优解。
3. 以自底向上或自顶向下的记忆化方式（备忘录法）计算出最优值
4. 根据计算最优值时得到的信息，构造问题的最优解

# 算法设计

动态规划的主要难点在于理论上的设计，也就是上面 4 个步骤的确定，一旦设计完成，实现部分就会非常简单。

使用动态规划求解问题，最重要的就是确定动态规划三要素：

1. 问题的阶段
2. 每个阶段的状态
3. 从前一个阶段转化到后一个阶段之间的递推关系。

递推关系必须是从次小的问题开始到较大的问题之间的转化，从这个角度来说，动态规划往往可以用递归程序来实现，不过因为递推可以充分利用前面保存的子问题的解来减少重复计算，所以对于大规模问题来说，有递归不可比拟的优势，这也是动态规划算法的核心之处。

确定了动态规划的这三要素，整个求解过程就可以用一个最优决策表来描述，最优决策表是一个二维表，其中行表示决策的阶段，列表示问题状态，表格需要填写的数据一般对应此问题的在某个阶段某个状态下的最优值（如最短路径，最长公共子序列，最大价值等），填表的过程就是根据递推关系，从 1 行 1 列开始，以行或者列优先的顺序，依次填写表格，最后根据整个表格的数据通过简单的取舍或者运算求得问题的最优解。

```
f(n,m)=max{f(n-1,m), f(n-1,m-w[n])+P(n,m)}
```

# 算法基本框架

```
for(j=1; j<=m; j=j+1) // 第一个阶段
   xn[j] = 初始值;

 for(i=n-1; i>=1; i=i-1)// 其他n-1个阶段
   for(j=1; j>=f(i); j=j+1)//f(i)与i有关的表达式
     xi[j]=j=max (或min) {g(xi-1[j1:j2]), ......, g(xi-1[jk:jk+1])};

t = g(x1[j1:j2]); // 由子问题的最优解求解整个问题的最优解的方案

print(x1[j1]);

for(i=2; i<=n-1; i=i+1)
{
     t = t-xi-1[ji];

     for(j=1; j>=f(i); j=j+1)
        if(t=xi[ji])
             break;
}
```