

# 漫谈五大常用算法

数据结构和算法

数据结构和算法

发布日期: 2018-08-12

阅读次数: 128

五大常用算法:

19.12.26 已整理

1. 分治
2. 动态规划
3. 贪心
4. 回溯
5. 分支界定

## 分治算法

### 概念

分治，就是“分而治之”，把一个问题分成两个或多个相同或相似的子问题，再把子问题分成更小的子问题.....最后子问题可以直接求解，原问题的解即子问题的解的合并。

分治法常常跟递归一起使用，借助递归，我们可以方便地将问题分解再将结果合并。

### 分治法的基本步骤

1. 分解：将原问题分解为若干个规模较小，相互独立，与原问题形式相同的子问题；
2. 递归：若子问题规模较小而容易被解决则直接解，否则递归地解各个子问题
3. 合并：将各个子问题的解合并为原问题的解。

## 可使用分治法求解的经典问题

1. 二分搜索
  2. 大整数除法
  3. Strassen矩阵乘法
  4. 棋盘覆盖
  5. [归并排序](#)
  6. [快速排序](#)
  7. 线性时间选择
  8. 最接近点对问题
  9. 循环赛日程表
  10. 汉诺塔
- 

## 动态规划

### 概念

将一个问题分解成若干个子问题（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。

### 与分治法的差别

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）。

## 动态规划的基本步骤

动态规划过程如下：

初始状态  $\rightarrow$  | 决策 1 |  $\rightarrow$  | 决策 2 |  $\rightarrow$  .....  $\rightarrow$  | 决策 n |  $\rightarrow$  结束状态

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：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。

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。

## 贪心算法

### 基本概念

在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的仅是在某种意义上的局部最优解。

贪心算法不是对所有问题都能得到整体最优解，选择的贪心策略必须具备无后效性，即某个状态以后的过程不会影响以前的状态，只与当前状态有关。

### 适用前提

局部最优策略能导致产生全局最优解。

### 贪心算法基本步骤

1. 建立数学模型来描述问题。
2. 把求解的问题分成若干个子问题。
3. 对每一子问题求解，得到子问题的局部最优解。
4. 把子问题的解局部最优解合成原来解问题的一个解。

# 回溯法

## 概念

在搜索尝试过程中寻找问题的解，当发现已不满足求解条件时，就“回溯”返回，尝试别的路径。

回溯法是一种选优搜索法，按选优条件向前搜索，以达到目标。但当探索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择，这种走不通就退回再走的技术为回溯法，而满足回溯条件的某个状态的点称为“回溯点”。

## 基本思想

在包含问题的所有解的解空间树中，按照深度优先搜索的策略，从根结点出发深度探索解空间树。当探索到某一结点时，要先判断该结点是否包含问题的解，如果包含，就从该结点出发继续探索下去，如果该结点不包含问题的解，则逐层向其祖先结点回溯。

（其实回溯法就是对隐式图的深度优先搜索算法）。

若用回溯法求问题的所有解时，要回溯到根，且根结点的所有可行的子树都要已被搜索遍才结束。

而若使用回溯法求任一个解时，只要搜索到问题的一个解就可以结束。

## 回溯基本步骤

1. 针对所给问题，确定问题的解空间:首先应明确定义问题的解空间，问题的解空间应至少包含问题的一个（最优）解；
2. 确定结点的扩展搜索规则；
3. 以 深度优先 方式搜索解空间，并在搜索过程中用剪枝函数避免无效搜索。

## 分支限界法

回溯法的求解目标是找出T中满足约束条件的所有解，而分支限界法的求解目标则是找出满足约束条件的一个解，或是在满足约束条件的解中找出使某一目标函数值达到极大或极小的解，即在某种意义下的最优解。

## 分支搜索

所谓“分支”就是采用 广度优先 的策略，依次搜索E-结点的所有分支，也就是所有相邻结点，抛弃不满足约束条件的结点，其余结点加入活结点表。然后从表中选择一个结点作为下一个E-结点，继续搜索。