# 原本

1. A

算法可以有0个或多个输入，但至少要有一个输出

1. C

操作性最好且最具有实用价值的是最坏情况下的时间复杂性

1. D

C = F（N I A），复杂性由问题规模、输入规模、采用算法三方面决定

1. D

动规的实质在于分而治之和解决冗余

和动规的主要区别在于，动规的子问题不独立，要依赖于已经解决的子问题

1. C

排序用nlogn，遍历统计用n，所以总共O(nlogn)

1. B

贪心选择解决背包问题，0-1背包用动规。0-1背包无法用贪心，因为贪心无法保证最终把背包装满，闲置的背包空间降低了单位背包空间的价值

1. A

1-(1-0.5)^2

1. A

快排最优的情况是用于划分的元素能把数组分的很均匀，A里随便挑哪一个都能划分的很均匀

1. C

TSP是排列树，A是子集树（完全八叉树），B是子集树（完全m叉树），D是子集树（完全四叉树）

回溯法一章中的实例有：装载问题、**批作业处理问题**、符号三角形问题、n后问题、0-1背包问题、最大团问题、图的m着色问题、**旅行售货员（TSP）问题**、**圆排列问题**、**电路板排列问题**、**连续邮资问题**

其中，批作业处理、TSP、圆排列、电路板排列、连续邮资是排列树，其余的都是子集树

1. D

如果要找可行解，回溯法的深度优先找到一条能到达叶子结点的路径就行，所以找可行解通常用回溯法

1. C

不一定会被弹出并分析，比如分支限界法一章中单源最短路径问题中，如果两个结点被同一个结点控制，那么分析其中一个结点，有可能直接就把另一个结点对应的子树减去，另一个结点就不需要弹出分析（见算法分析-5.ppt的第58页）

1. B

拉斯维加斯算法的特征：得到解的一定是正确解，但有时候得不到解

舍伍德算法可以对确定算法进行改造，但只是为了消除或降低好坏实例之间的复杂性差别、优化平均性能，说降低复杂性也有些不妥

1. A

LCS填空

基本思路：

如果X和Y的最后一个元素相等，那么最长子序列就是Xm-1和Yn-1的最长公共子序列长度再加上最后一个元素，所以是c[i][j]=c[i-1][j-1]+1

如果最后一个元素不相等，那么应该先找到Xm-1、Yn的最长公共子序列和Xm、Yn-1的最长公共子序列，比较二者，较长的就是当前的最长公共子序列

（6）C

（7）A

（8）感觉这题有点问题，题目应该是子集树，ACD都是子集树

（9）D

（10）A

（11）B

（12）H

分析阶关系：

（2）Ω

（3）θ（？）

（4）O

2.

相同：都需要在求解之前定义和组织解空间、都是通过在解空间搜索来寻找问题的解

不同：

搜索方式上，回溯法是深度优先，分支限界法是广度优先

搜索策略上，回溯法是根据剪枝函数，选择下一个扩展结点并按深度优先的方式进行搜索

分支限界是在扩展结点处先产生所有子结点，然后（略）

1. O(nlogn)

# 自己整理

下面哪种排序的平均比较次数最少（D）

A插入排序

B选择排序

C堆排序

D快速排序

快速排序算法的性能取决于划分的对称性