算法设计与分析复习资料

一、基本概念与知识点，考查形式主要为选择题和填空题（50分左右）

（1）算法定义与特征

例：算法是由若干条指令组成的有穷序列，需要满足输入、输出、确定性、有效性和有限性。

（2）算法效率度量方法

例：假设算法A理论上的时间复杂度T(n)=O(n2)，现在有两台同类型计算机M1和M2，M2的计算速度是M1的x倍。若在M1和M2上分别测试算法A，则在相同时间内，M1和M2能够求解的问题规模n1和n2的关系。求解相同规模，两台机器耗费时间关系。

（3）对于分治法求解问题的算法复杂性理论分析

，对于给定a,b和*f*(n)的不同情况,判断算法时间复杂度。

例：对于某一具体问题，若*f*(*n*) = *O*(1)且*a=*2*，* *b=*2，则求解该问题的时间复杂度为*O*(n)

（4）课堂所讲授的分治、动态规划、贪心、回溯、分支限界、遗传等算法基本思想，求解问题步骤，关键点，时间/空间复杂度分析结论。不同算法适合求解的经典问题。

例：在无序序列中查找最大值，若采用分治法查找，则求解该问题的算法时间复杂度为（）

适合采用动态规划算法求解的代表性问题是（）

回溯法求解旅行售货员问题时的解空间树是什么树

适合采用动态规划算法或贪心算法求解的问题具备的共同特征是

求解以下问题的动态规划算法中，空间复杂度最小的是

动态规划算法求解问题步骤：最优子结构性质、递推定义最优解（递推方程）、自底向上求解最优值、构造最优解。

常见算法的递推方程：归并排序递推方程为；在一个给定的无序序列中查找固定值x，若采用分治法查找，算法的递推方程为。

回溯法一般按照 策略搜索解空间树，根据问题不同，解空间树一般可分为子集树和排列树两种结构，01背包问题的解空间树是 结构，旅行售货员TSP问题的解空间树是 结构。

分支限界法求解极大或极小问题剪枝方法。上界或下界 界函数

回溯法在解空间中搜索解的过程。

随机化算法中， 算法思想主要是解决不同输入实例之间的差异，性；符合猜的次数越多，猜对的概论越大是 算法核心思想。

二、算法设计与分析（三道题，30分左右）

1、给定算法伪代码描述，阅读理解算法，给出算法求解的问题和输出结果，分析算法的时间复杂性（有过程）。

2、给定问题描述，设计有效算法求解问题，并对所设计算法进行复杂度分析。重点：动态规划递推方程建立。矩阵链、最长公共子序列、背包、多阶段决策。

例：

1、算法伪代码阅读与分析

ALG(C, n)

IF *n*=1 THEN RETURN C[1]

ELSE

*temp*=ALG(C, *n*-1)

IF *temp*≤C[*n*] THEN RETURN *temp*

ELSE RETURN C[*n*]



（1）说明该算法的功能。

（2）该算法基本操作执行次数与相同规模下不同输入是否相关？

（3）分析该算法的时间复杂度。

2、算法设计

例：

已知一个国家有n个城市，已知每个城市过去一年的GDP总量，请找出其中GDP总量最为接近的两个城市。

（1）用数学方法描述该问题求解目标。

（2）为该问题设计一个有效算法，需要给出算法思想和伪代码。

float getMinDiff(a)

输入：数组a[0,n-1]

返回：最小差minDiff

sort(a)//按GDP非递减排序

minDiff ← a[n-1]

for i←1 to n-1

min ← a[i] – a[i-1]

if min < minDiff

minDiff ← min

return minDiff

（3）对你所设计的算法进行时间复杂度分析

在不考虑排序情况下：

**问题规模参数**为输入数组中元素个数；

**算法基本操作**为min ← a[i] – a[i-1]；

**影响基本操作执行次数的因素**只取决于问题规模n；

基本操作执行次数为n-1次；**时间复杂度为*O*(n)**。

排序可以在*O*(nlogn)时间复杂度下完成，因此**算法时间复杂度为取决于排序时间复杂度，为*O*(nlogn)**。

三、算法设计与实现（30分左右）

分治、动态规划、回溯法或分支限界法求解问题的表示，程序基本结构，伪代码描述，算法复杂性分析。

例：

设某一餐馆需要进行套餐设计，套餐由n种食物组成，每种食物都可以从m个不同的供应商处购得。设wij是从供应商j处采购食物i的质量，cij是相应的价格。现在希望得到一个总价格不超过d的最高质量套餐设计方案

分析该问题适合采用何种算法求解

该问题与课程中讨论的哪一个典型问题属于同一类问题

给出问题解的形式

给出求解问题的核心代码框架