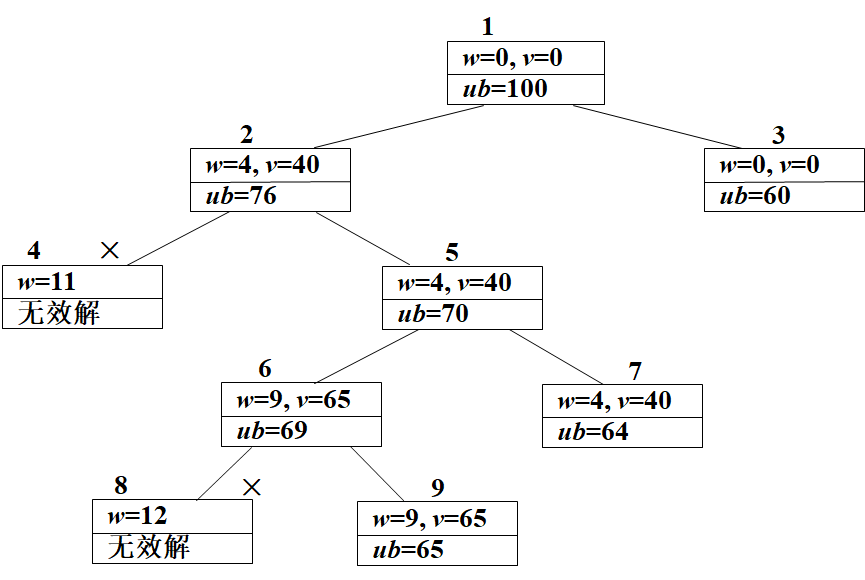
**0/1背包问题**：给定n个重量为｛w1，w2，...，wn｝、价值为｛v1，v2，...，vn｝的物品和一个容量为C的背包，应选择哪些物品装入背包，才能使装入背包的物品价值最高？

蛮力法：给出所有子集，计算子集的总重量和总价值，进行比较。

动态规划法：证明0/1背包问题，满足最优性原理，

分支限界法：用贪心法求得背包问题的下界，再求得上界：将背包中剩余容量全部装入第i+1个物品，并可以将背包装满，限界函数：*ub=v+(W-w)\*(vi+1/wi+1)*。

总结：

1.剪枝函数给出每个可行结点相应的子树可能获得的最大价值的上界。

2.如这个上界不会比当前最优值更大，则可以剪去相应的子树。

3.也可将上界函数确定的每个结点的上界值作为优先级，以该优先级的非增序抽取当前扩展结点。由此可快速获得最优解。

贪心法：选择单位重量价值最大的物品。

**哈密顿回路问题**：共有n个城市，要求从一个城市出发，经过每个城市恰好一次，最后回到出发城市。

蛮力法：对于给定的无向图G=（V，E），依次考察图中所有顶点的全排列，满足以下条件的全排列（vi1,vi2,...,vin）构成的回路就是哈密顿回路：（1）相邻顶点之间存在边，即（vij,vij+1）∈E（1≤j≤n-1）（2）最后一个顶点和第一个顶点之间存在边，即（vin,vi1）∈E

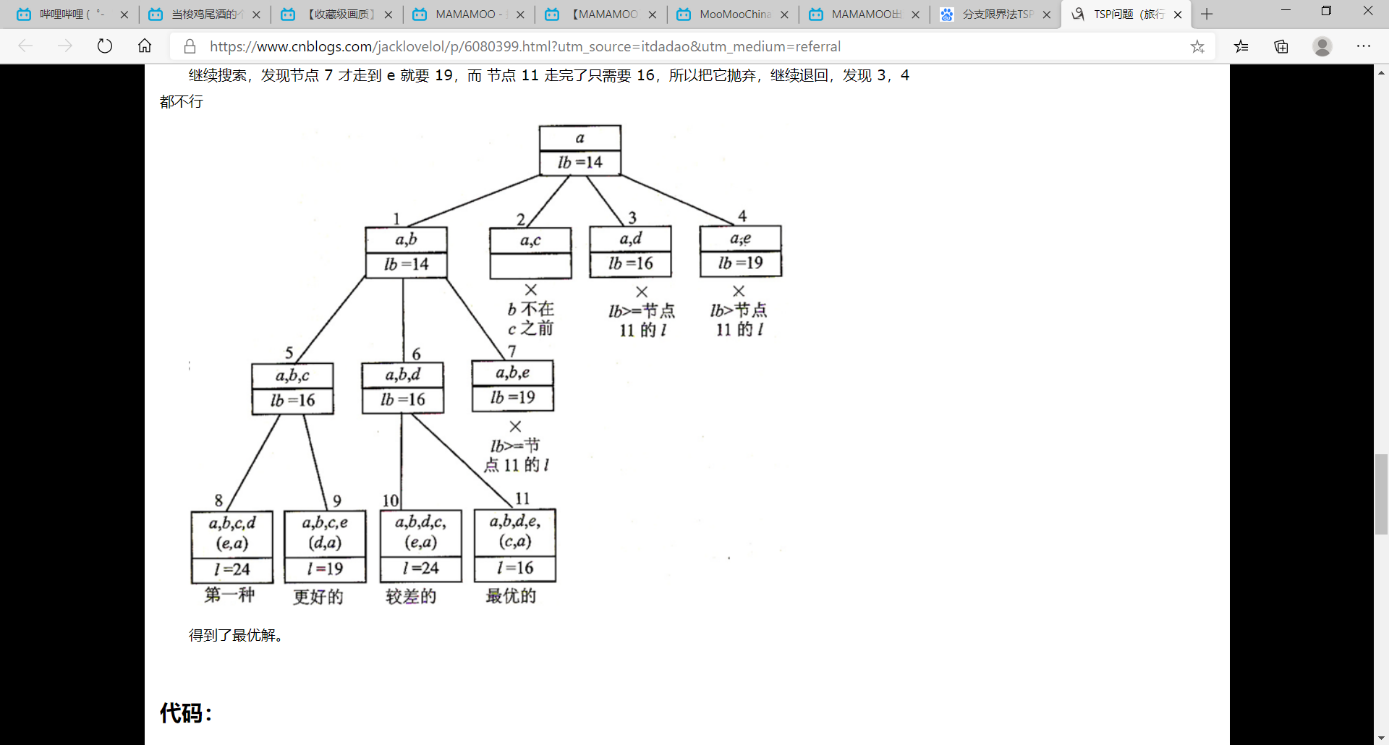
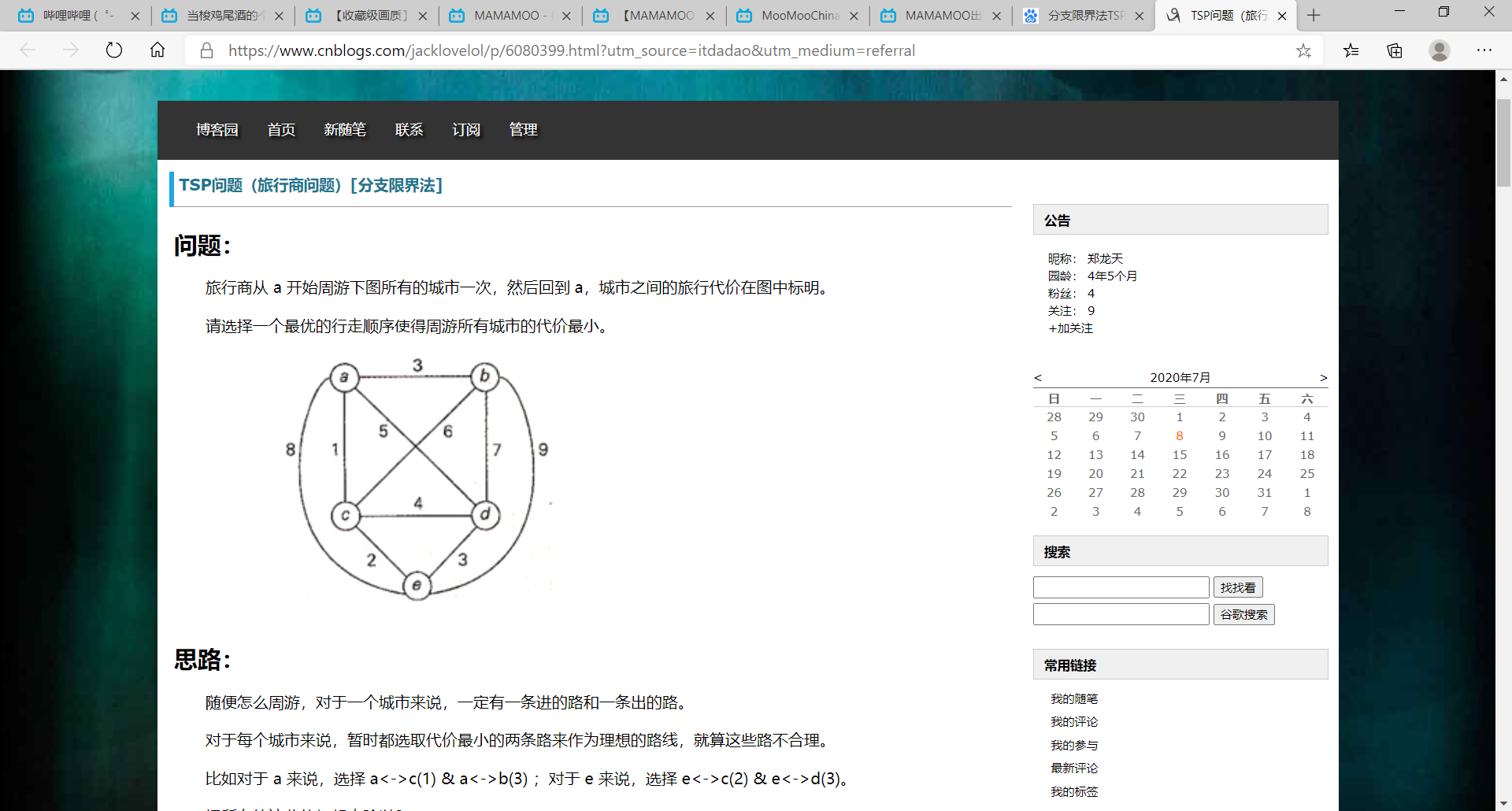
回溯法：假定图G=（V，E）的顶点集为V=｛1,2,…,n｝，则哈密顿回路的可能解表示为n元组X=（x1,x2,…,xn），其中，xi｛1,2,…,n｝。根据题意，有如下约束条件：

首先把所有顶点的访问标志初始化为0，然后依次为每个顶点着色。在解空间树中，如果从根结点到当前结点对应一个部分解，即满足上述约束条件，则在当前结点处选择第一棵子树继续搜索，否则，对当前子树的兄弟子树继续搜索，即为当前顶点着下一个颜色。

**TSP问题**：旅行家要旅行n个城市然后回到出发城市，要求每个城市经历且仅经历一次，并要求所走的路程最短。

蛮力法：找出所有可能的旅行路线，即依次考察所有顶点的全排列，从中选出路径长度最短的哈密顿回路。

分支限界法：通过贪心法获得上界，将矩阵中每一行最小的两个元素相加除以2，再对这个结果取整，得到下界，

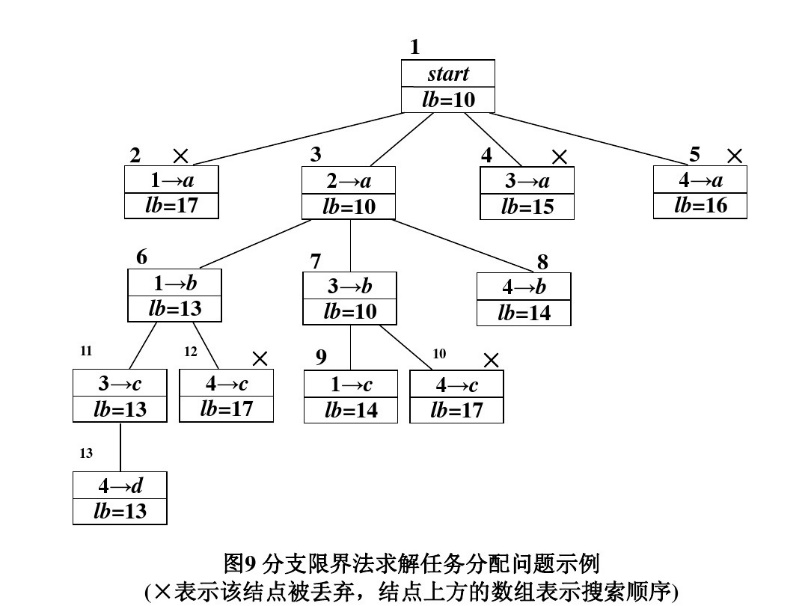


动态规划法：确定TSP问题满足最优性原理，d( k,V’-{k} )表示从顶点k出发经过V’-{k}中各个顶点一次且仅一次，最后回到出发点i的最短路径长度，则：

贪心法：1.最近邻点策略：从任意城市出发，每次在没有到过到的城市中选择最近的一个，直到经过了所有的城市，最后回到出发城市。2.最短链接策略：每次在整个图的范围内选择最短边加入解集合中，但是要保证加入解集合中的边最终形成一个哈密顿回路。

**任务分配问题**：假设有n个任务需要分配给n个人执行，每个任务只分配给一个人，每个人只执行一个任务，且第i个人执行第j个任务的成本是Cij（1≤i,j≤n），找出总成本最小的方案。

蛮力法：用矩阵表示任务分配问题的成本，在成本矩阵中的每一行选取一个元素，这些元素分别属于不同的列，且元素之和最小。

分支限界法：用贪心法求得上界，将每人的最小成本加起来形成下界，如图对解空间树进行搜索，结点13对应的解即为最优解，搜索结束。为求得最优解的各个分量，从结点13开始回溯，得到最优解

任务1任务2任务3任务4

**图着色问题**：给定无向联通图G=（V，E），求图G的最小色数k，使得k种颜色对G中的顶点着色，可使任意两个相邻顶点着不同颜色。

贪心法：选择一种颜色，用该颜色为尽可能多的顶点着色，例如，选取颜色1，依次考察图中的未被着色的每个顶点，如果该顶点的邻接点都还未被着色，则用颜色1为该顶点着色。

回溯法：首先把所有顶点的颜色初始化为0，然后依次为每个顶点着色。在解空间树中，如果从根结点到当前结点对应一个部分解，即所有的颜色指派都没有冲突，则在当前结点处选择第一棵子树继续搜索即为下一个顶点着颜色1，否则，对当前子树的兄弟子树继续搜索，即为当前顶点着下一个颜色。

**批处理作业问题**：n个作业{1,2,…,n}要在两台机器上处理，每个作业必须先由机器1处理，然后再由机器2处理，机器1处理作业i所需时间为ai，机器2处理作业i所需时间为bi(1≤i≤n)，要求却盯着n个作业的最优处理顺序，使得第1个作业在机器1上处理开始，到最后一个作业在机器2上处理结束所需时间最少。

回溯法：使机器1没有空闲时间，且机器2的空闲时间最小

sum1[n], sum2[n]保存在调度过程中机器1和机器2的当前完成时间

sum1[k]=sum1[k-1]+作业x[k]在机器1的处理时间

sum2[k]=max{sum1[k],sum2[k-1]}+作业x[k]在机器2的处理时间

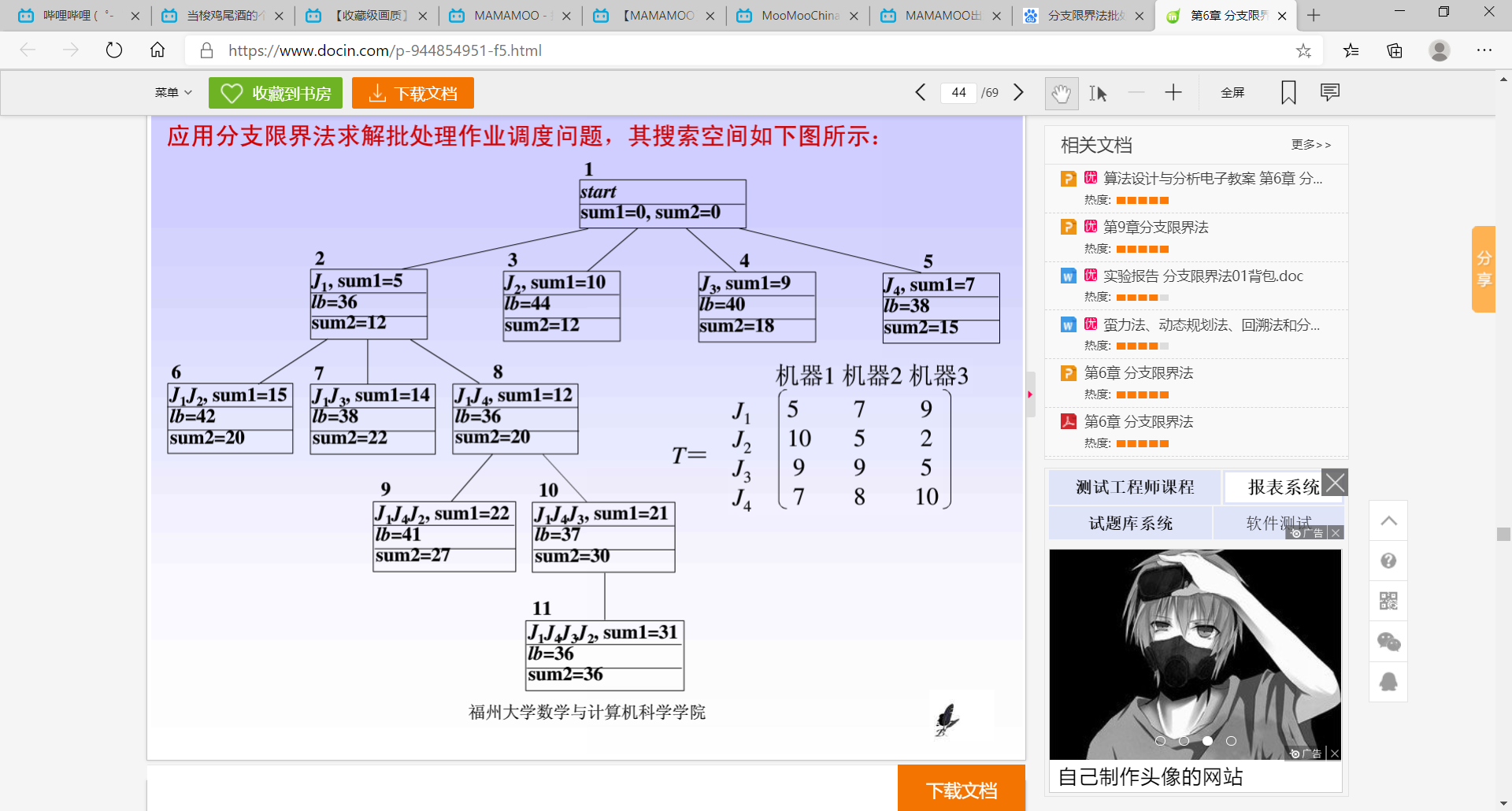
分支限界法：随机产生几个调度方案，从中选取具有最短完成时间的调度方案作为近似最优解，作为批处理作业调度问题的上界，关键在于如何估算部分解的下界

sum1, sum2表示机器1和机器2完成k个作业的处理时间，现在要处理作业k+1

sum1=sum1+tk+1,1

lb=max{sum1,sum2}++min{}

sum2=max{sum1,sum2}+



**蛮力法**（又称穷举法）所依赖的基本技术使遍历，即采用一定策略依次处理待求解问题的所有元素，从而找出问题的解。

查找问题中的：顺序查找（P36）、串匹配问题（P37）

排序问题中的：选择排序（P42）、起泡排序（P43）

组合问题中的：0/1背包问题（P44）、任务分配问题（P45）

图问题中的：哈密顿回路问题（P46）、TSP问题（P47）

几何问题中的：最近对问题（P48）、凸包问题（P49）

**分治法**，将一个难以直接解决的大问题划分成一些规模较小的子问题，分别求解各个子问题，再合并子问题的解得到原问题的解。

排序问题中的：归并排序（P59）、快速排序（P61）

组合问题中的：最大子段和问题（P64）、棋盘覆盖问题（P65）

几何问题中的：最近对问题（P68）、凸包问题（P71）

减治法将原问题分解成若干个子问题，并且将原问题（规模为n）的解与子问题（规模通常是n/2或n-1）的解之间存在某种确定的关系，通常为①原问题的解只存在于其中一个较小规模的子问题中；②原问题的解与其中一个较小规模的解之间存在某种对应关系。

查找问题中的：折半查找（P80）、二叉查找树（P82）、选择问题（P84）

排序问题中的：插入排序（P86）、堆排序（P88）

组合问题中的：淘汰赛冠军问题（P90）、假币问题（P91）

**动态规划法**，求解过程分3个阶段：①划分子问题将待求解问题分解成若干个相互重叠的子问题，每个子问题对应一个决策阶段，并且子问题之间具有重叠关系；②确定动态规划函数：根据子问题之间的重叠关系找到子问题满足的递推关系式；③填写表格：设计表格，以自底向上的方式计算各个子问题的解并填表，实现动态规划过程。

图问题中的：多段图的最短路径问题（P102）、多源点最短路径问题（P106）、TSP问题（P107）

组合问题中的：最长递增子序列问题（P109）、最长公共子序列问题（P111）、0/1背包问题（P114）

查找问题中的：最优二叉查找树（P116）、近似串匹配问题（P119）

**贪心法**把一个复杂问题分解为一系列较为简单的局部最优选择，每一步选择都是对当前解的一个扩展，直到获得问题的完整解。，一般能获得近似最优解，不总能获得整体最优解。

图问题中的：TSP问题（P129）、图着色问题（P132）、最小生成树问题（P134）

组合问题中的：背包问题（P138）、活动安排问题（P140）、多机调度问题（P142）

**回溯法**从解空间树的根结点出发，按照深度优先策略搜索满足约束条件的解。在搜索至树中某结点时，先判断该结点对应的部分解是否满足约束条件，即判断该结点是否包含问题的最优解，如果肯定不包含，则跳过以该结点为根的子树即剪枝；否则进入以该结点为根的子树，继续按照深度优先策略进行搜索。

图问题中的：图着色问题（P155）、哈密顿回路（P158）

组合问题中的：八皇后问题（P160）、批作业处理问题（P163）

分支限界法首先确定一个合理的限界函数，并根据限界函数确定目标函数的界。然后按照广度优先策略搜索问题的解空间树，在分支结点上，依次扩展该结点的所有孩子结点，分别估算这些孩子结点的目标函数的可能取值，如果某孩子结点的目标函数的可能取值超出目标函数的界，则将其丢弃，否则将其加入待处理结点表中。依次从结点表中选取使目标函数取得极值的结点成为当前扩展结点，重复上述过程，直至找到最优解。

图问题中的：TSP问题（P175）、多段图的最短路径问题（P178）

组合问题中的：0/1背包问题（P180）、任务分配问题（P182）、批处理作业问题（P184）