《算法设计与分析》第5次作业答案

姓名：XXX　　　　学号：XXXXXXXX

算法分析题

题目1：请论述回溯法和分支限界法的相同点与不同点。

答：相同点：都是在问题的解空间树 T 上搜索问题解的算法。

不同点：

1、求解目标不同：回溯法的求解目标是找出解空间中满足约束条件的所有解；分支限界法的求解目标是找出满足约束条件的一个解或特定意义下的最优解。

2、对解空间树的搜索方式不同：回溯法进行深度优先搜索；分支限界法进行广度优先或最小消耗优先搜索。

3、存储结点的常用数据结构不同：回溯法采用堆栈；分支限界法采用队列、优先队列。

4、遍历所需的空间不同：回溯法树的高度；分支限界法队列的长度。

5、搜索用到的函数不同：回溯法-约束函数、限界函数；分支限界法-约束函数、限界函数、优先级函数。

题目2：设某一机器由n个部件组成，每一种部件都可以从m个不同的供应商处购得，设W_{ij}是从供应商j 处购得的部件i 的重量，C_{ij} 是相应的价格，试设计一个回溯法，给出总价格不超过d 的最小重量机器设计。请描述算法的基本思想，要求画出解空间树，并给出相应的剪枝条件。试通过下面这个例子进行说明。

例子：假设$n = 3$，$m = 3$，机器部件重量w_{ij} 和价格c_{ij} 分别如下表所示，$d = 15$。

w_{ij}	j=1	j=2	j=3
i=1	4	2	8
i=2	5	2	1
i=3	2	2	3

c_{ij}	j=1	j=2	j=3
i=1	10	6	12
i=2	8	9	5
i=3	2	5	4

答：

算法思路：初始化当前价格$cp = 0$，当前重量$cw = 0$，此外设置一个变量sum 表示选择部件的总重量。循环选择第i 种部件时，判断从j 号供应商购买部件后的价格是否大于d，如果不大于则选择，否则不选。然后继续选择下一供应商进行判断。在得到一种部件的合适的供应商后，继续选择下一部件的供应商，从第一个选到最后一个供应商。当所有部件选择结束后，把得到的总重量与当前 sum比较，如果小就赋给sum，然后从这一步开始，回溯

到上一部件选择下一合适供应商，继续搜索可行解，直到将整个解空间树搜索完毕。这样的得到的sum 即为最优解。考虑到时间复杂度，加一个剪枝条件，即在每次选择某一部件时，再判断选择后的当前重量是否已经大于sum，如果大于就没必要继续搜索，剪枝即可。
例子说明：可行性约束函数为：当前已选好供应商的零件价格之和 + 当前未选好供应商的零件的最低价格之和 $\leqslant$ 15。
限界函数为：当前已选好供应商的零件重量之和 + 当前未选好供应商的零件最小重量之和 $\leqslant$ 当前最优解。

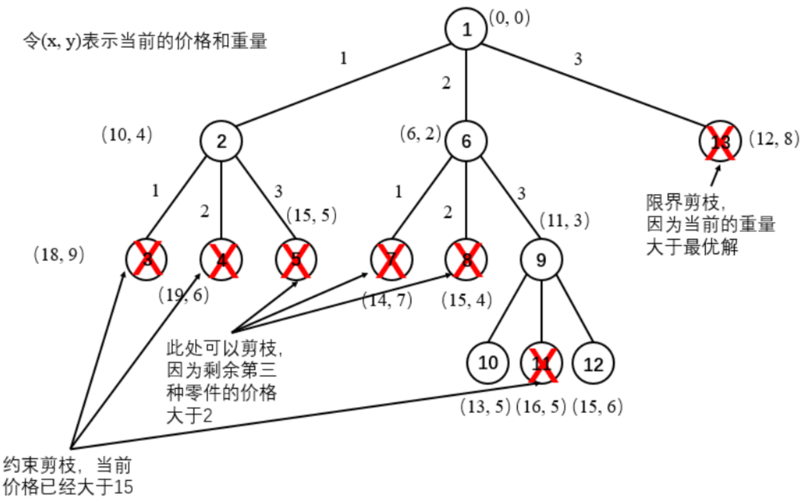

Figure 1: 回溯法解空间树

如上节点序号表示遍历的顺序。最优解为从供应商2 购买零件1，从供应商3 购买零件2，从供应商1 购买零件3。价格不超过15 的最小重量为5。

题目3：假设有 n 个任务由 k 个可并行工作的机器完成，完成任务i需要的时间为T_i，请使用分支限界法找出完成这 n 个任务的最佳调度，使得完成全部任务的时间最早。请描述算法思想，以$n = 7$，$k = 3$，$t[1..7] = [2, 14, 4, 16, 6, 5, 3]$为例，要求画出解空间树，并给出相应的剪枝条件。
答：
算法思想：可以将该问题的解空间树表示成一个k 叉树，树的第i 层表示第i 个任务的处理有k种机器可选择。假设当前最优解，即已经分配的任务中完成所需要的最长时间为R_{best}，将其初始为无限大。假设T_i^t表示第i个机器处理任务的总时间，其中$i \in [1, k]$，最优解是在最优调度策略的情况下，T_i^t中的最大值T_{max}。优先级队列的优先级由当前调度策略中对应的T_i^t的最大值的大小决定，最大值T_{max}越小的节点优先级越高。首先将$t[1..n]$降序排列，从运行时间最长的任务开始遍历。对解空间树做广度优先搜索，每次搜索时，根据优先级，选出一个代价函数值最优的结点，若不用对其剪枝，则生成其子节点并将子节点入队；若搜索到达叶子节点，则更新当前的解。搜索过程中对满足剪枝条件的节点进行剪枝。
剪枝条件：当前节点下，T_i^t中的最大值T_{max}大于R_{best}
例子说明：首先对t做降序排序为t[1..7] = [16, 14, 6, 5, 4, 3, 2]

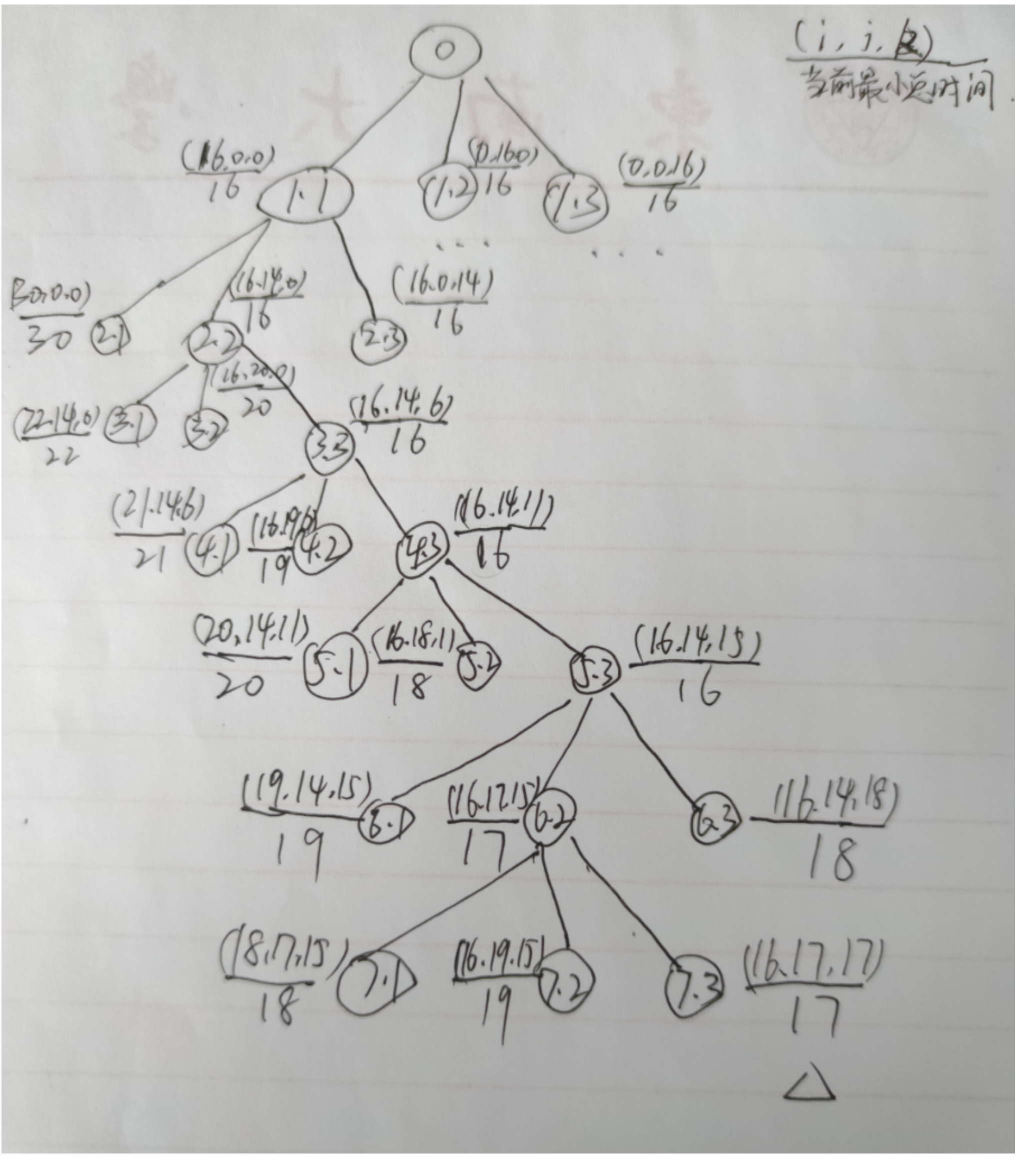

Figure 2: 分支限界法解空间树(图源自某位同学)

从根节点出发开始广度优先搜索，遍历前三个节点后可以得到T_{max}为16，假设当前的解向量是$< 1,2,3 >$，即任务1,2,3分别被分配到机器1,2,3上面。然后根据优先级函数以及剪枝条件，可以判断任务4和任务5都应该放在机器3上面，此时的T_{max}为16。再根据优先级函数以及剪枝条件，当任务6放在机器2上执行时，T_{max}最小为17(因为放在机器1上是19，放在机器3上是18)，最后搜索到叶子节点也就是第7个任务，根据优先级函数以及剪枝条件，可以判断任务7应该放在机器3上执行。此时得到一个最优的解向量为$< 1,2,3,3,3,2,3 >$同理，当最初三个任务的解向量是$< 1,3,2 >$等等其他组合时，也有对应的最优解向量$< 1,3,2,2,2,3,2 >$等等。综上所述。这些调度的三台机器的时间分配均为16，17，17，最早完成时间均为17。