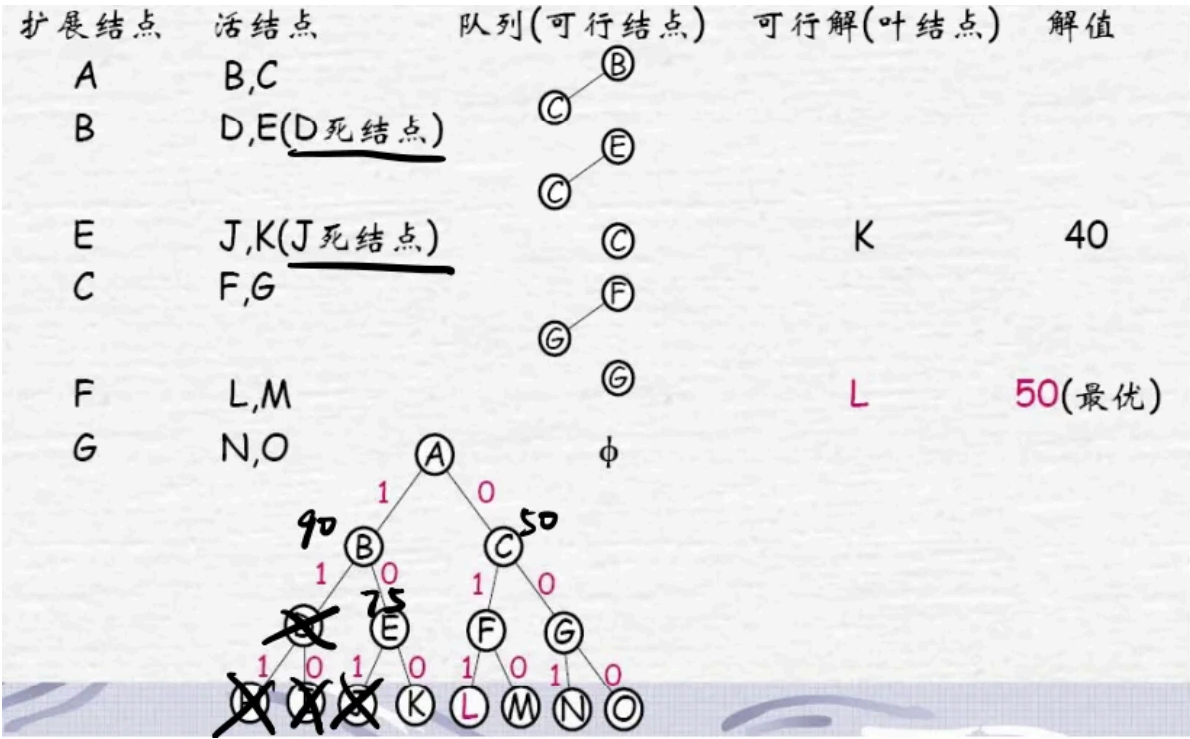
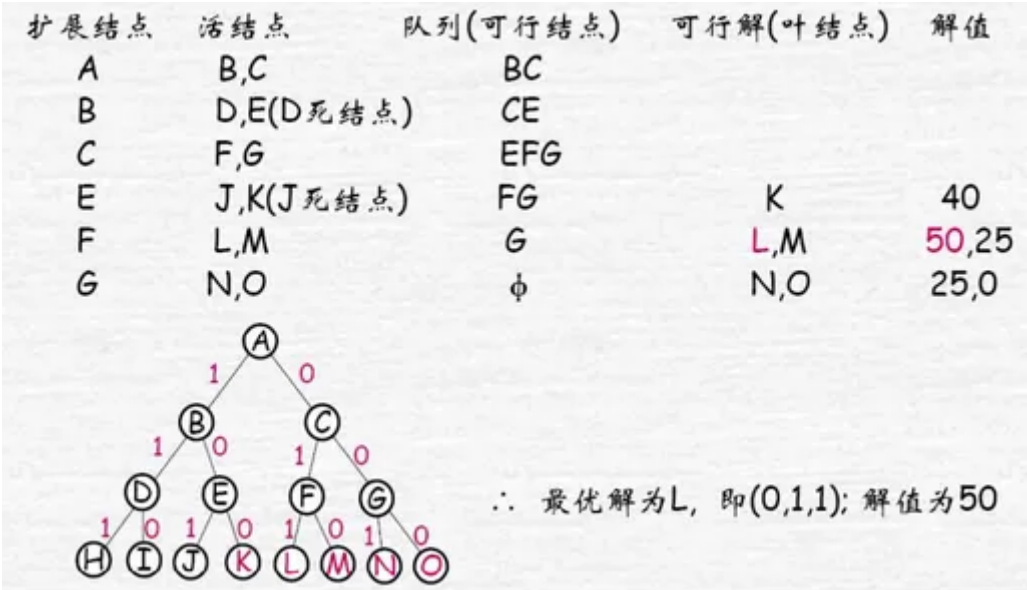


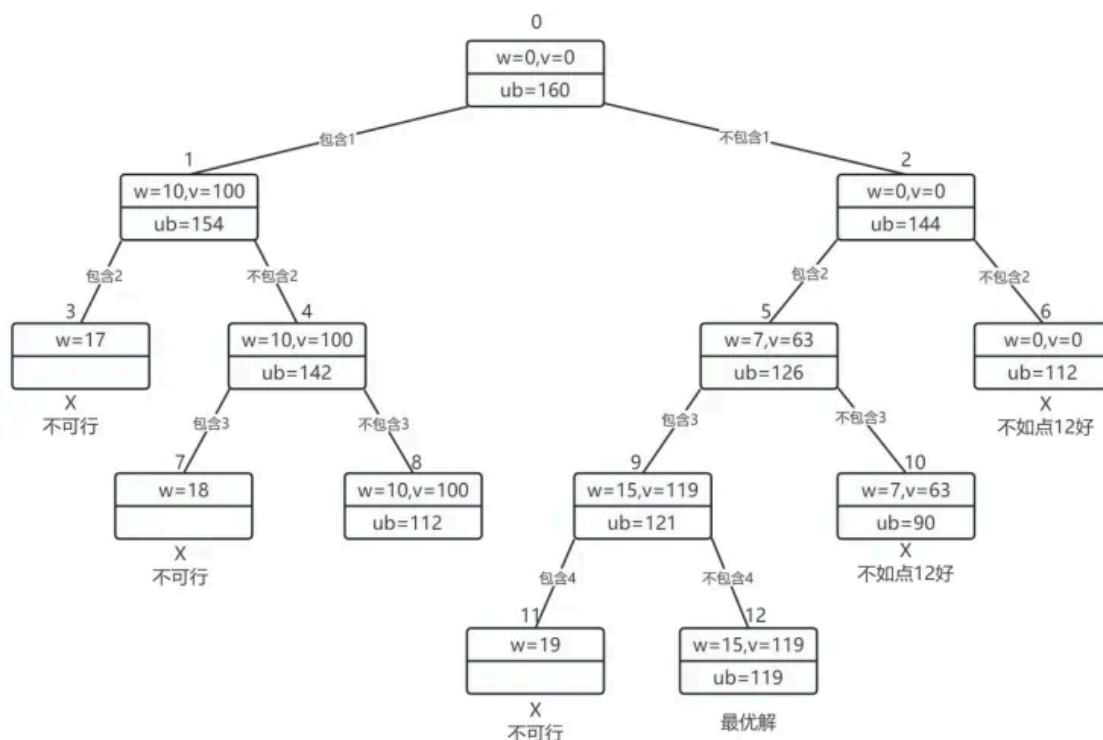
分支限界法

- 分支限界法常以**广度优先**或以**最小耗费(最大效益)优先**的方式搜索问题的解空间树，裁剪那些不能得到最优解的子树以提高搜索效率。
- 步骤：① 定义解空间(对解编码)；② 确定解空间的树结构；③ 按BFS等方式搜索：
 - a. 每个活结点**仅有一次机会**变成扩展结点；
 - b. 由扩展结点生成一步可达的新结点；
 - c. 在新结点中，删除不可能导出最优解的结点；//限界策略
 - d. 将剩余的新结点加入活动表(队列)中；
 - e. 从活动表中选择结点再扩展；//分支策略
 - f. 直至活动表为空；
- 队列式FIFO分支限界
- 优先队列分支限界

0-1背包问题



物品	重量	价值/美元	价值/重量
1	10	100	10
2	7	63	9
3	8	56	7
4	4	12	3



装载问题

有一批共个集装箱要装上2艘载重量分别为 C_1 和 C_2 的轮船，其中集装箱 i 的重量为 W_i ，且 $\sum_{i=1}^n w_i \leq c_1 + c_2$

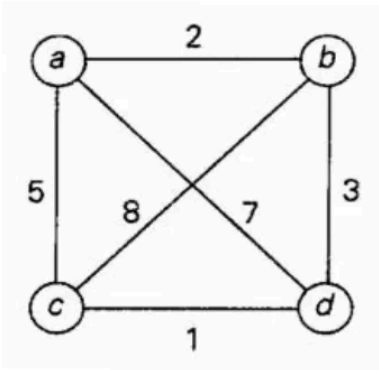
装载问题要求确定是否有一个合理的装载方案可将这个集装箱装上这2艘轮船。如果有，找出一种装载方案。

容易证明：如果一个给定装载问题有解，则采用下面的策略可得到最优装载方案。

- (1) 首先将第一艘轮船尽可能装满；
- (2) 将剩余的集装箱装上第二艘轮船。

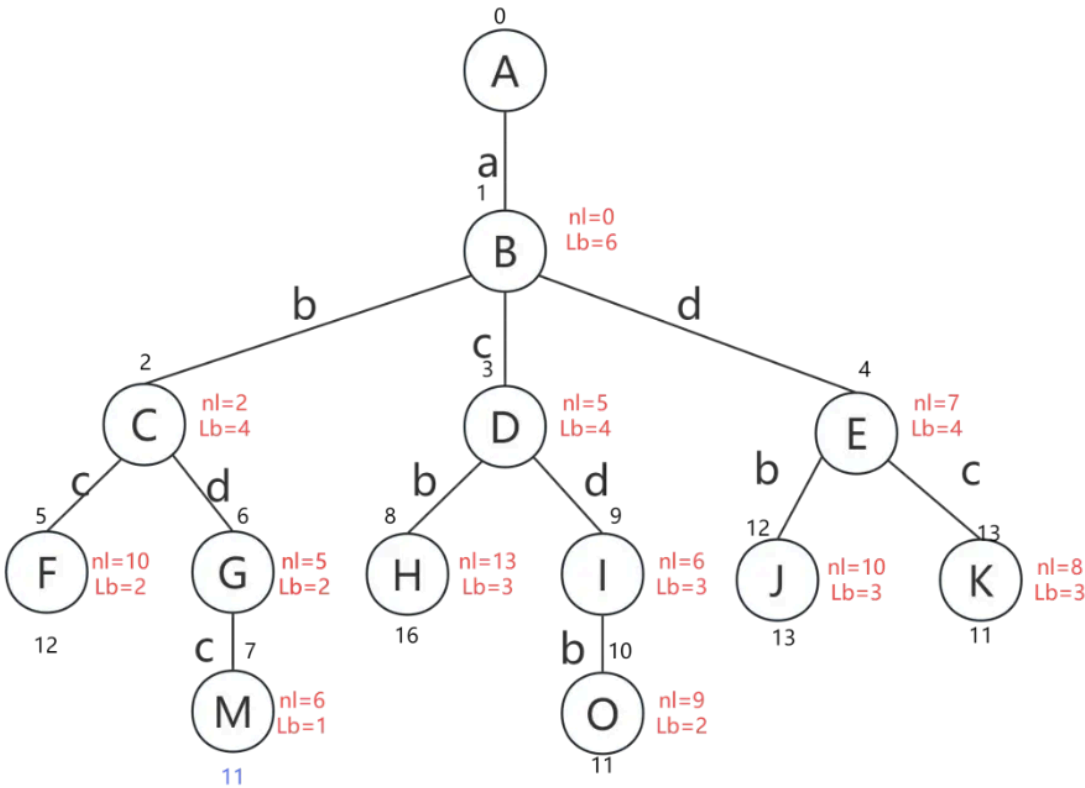
TSP问题

对于图 3，应用分支限界法求解从 a 点开始的 TSP 问题，请详细给出解空间树，搜索过程及最优解。



-1	2	5	7
2	-1	8	3
5	8	-1	1
7	3	1	-1

nl代表其当前所走路程的长度，Lb代表所有可行解的下界，即每一个节点的出边之和。B(0, 6)进队，其Lb=6的计算方式：找到邻接矩阵中每一行或者每一列除-1之外最小权值相加，即2+2+1+1=6。



回溯法与分支限界区别

回溯法与分支限界法

1. **求解目标不同**：一般而言，回溯法的求解目标是找出解空间树中满足约束条件的**所有解**，而分支限界法的求解目标则是尽快地找出满足约束条件的**一个解**；
1. **搜索方法不同**：回溯法使用**深度优先方法搜索**，而分支限界一般用**宽度优先或最佳优先方法**来搜索；
1. **对扩展结点的扩展方式不同**：分支限界法中，每一个活结点**只有一次机会成为扩展结点**。活结点一旦成为扩展结点，就一次性产生其所有儿子结点
1. **存储空间的要求不同**：分支限界法的**存储空间比回溯法大得多**，因此当内存容量有限时，回溯法成功的可能性更大。

回溯法与穷举法

穷举法：分解后检查。要将一个解的各个部分全部生成后，才检查是否满足条件，若不满足，则直接放弃该完整解，然后再尝试另一个可能的完整解，它并没有沿着一个可能的完整解的各个部分逐步回退生成解的过程。

回溯法：动态生成解空间。一个解的各个部分是逐步生成的，当发现当前生成的某部分不满足约束条件时，就放弃该步所做的工作，退到上一步进行新的尝试，而不是放弃整个解重来