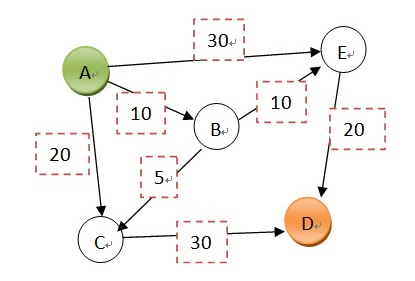
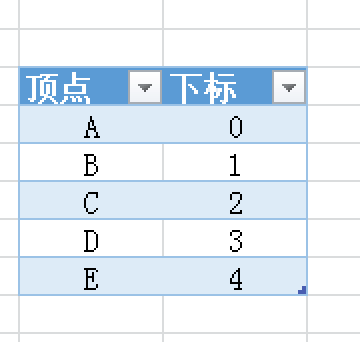
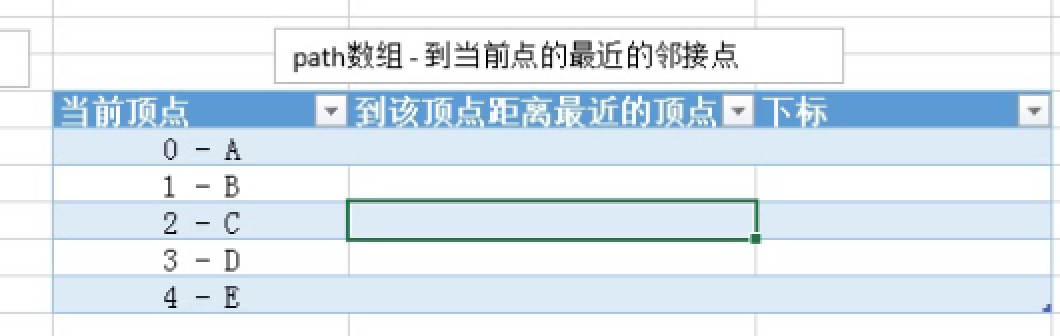
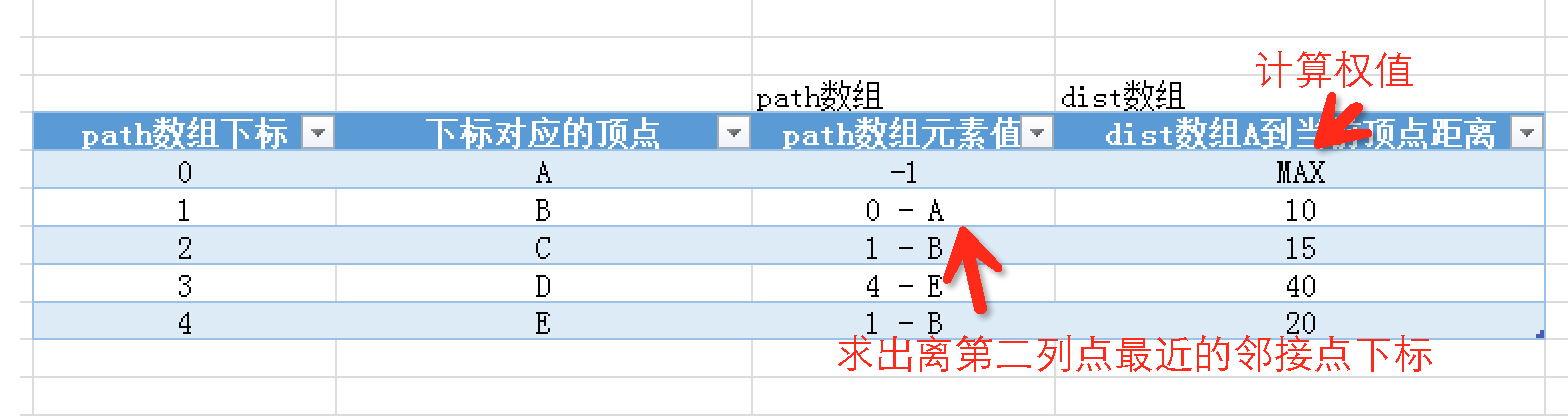
1. 最短路径
   1. 概念：从某一个点到另一个点，经过的路径的权值之和最小
   2. 
   3. 找A到D的最短路径 ABED
   4. 倒推出的特性，找到A到D最短，相当于找到 B到D最短，以及C到D最短
   5. 程序也是一样，需要倒推出最短路径，从D往回推
   6. 分析项目需求，求A到D最短路径
   7. 打开Excel分析
      1. 
      2. 顶点的名字
      3. 推导 从**A到E**的最短路径
      4. 从A出发，找A的邻接点 A-E A – B A – C
      5. A到E 是 30 A B E加起来是20，C到达不了，因此找A到E
   8. 交代程序是如果做的
      1. dist[]数组
      2. 
      3. 从出发点到各个点最短的距离
      4. path[]数组，当当前点最近的邻接点
      5. 
   9. 代码分析
      1. 先初始化所有数据
      2. 开始进入循环，遍历所有点的关系
      3. 更新最小点
   10. 再用Excel分析步骤
   11. 
   12. 在通过得到的数据求出最短路径
   13. 可以做演示
2. 二叉排序树
   1. 树不要研究 就研究 二叉树，二叉树的特例有个二叉排序树，特点，中序遍历后的结果是升序序列
   2. 在高一级的是平衡二叉树， 然后是红黑树，再就是红黑树变体
   3. 之所以研究平衡二叉树原因是，当二叉排序树删除后，可能退化为链表，因此需要做平衡
   4. 构造二叉排序树，从Excel中拿到数组，然后构造二叉排序树，画图
   5. 然后查看中序遍历结果，是升序的
   6. 规律：
      1. 新添加的节点作为叶子存在
3. 二叉排序树代码
   1. 讲解创建
   2. 讲解遍历
   3. 讲解查找
   4. 讲解查找极值
   5. 讲解删除
      1. 1、删除叶子 简单
      2. 2、删除只有左子树的或者只有右子树的 简单
      3. 3、删除有左有右子树的 复杂
         1. 有一种方式是重新构造一棵树，构造时候不插入要删除的节点，但是效率太低
         2. 用画图先进行中序遍历，找到直接前驱和后继
         3. 
         4. 代码中如果找直接前驱，就找左子树中最右的子树
         5. 直接后继是找右子树中最左的即可
         6. 找到直接前驱标记为S，删除节点标记为P，将S数值替换到P上，然后删除S即可
4. 平衡二叉树
   1. 先说明平衡二叉树概念以及平衡因子
   2. 如何维持平衡，利用节点的翻转
   3. 翻转的几种形式 LL RR LR RL