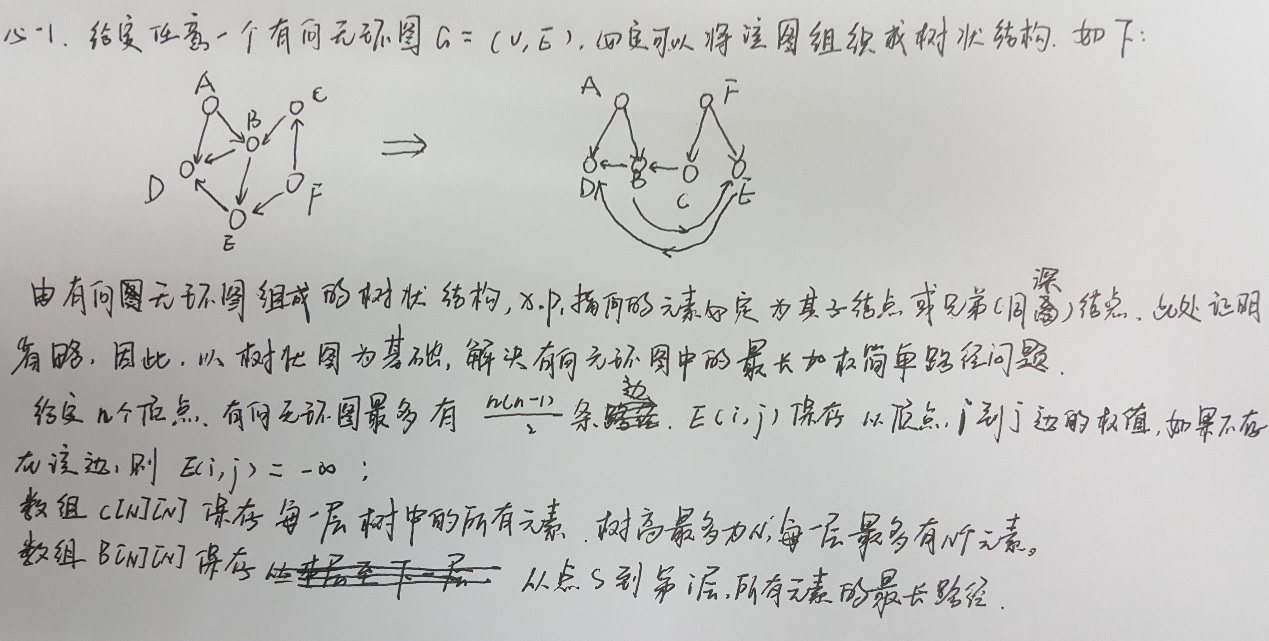
思考题15-1：



因此动态规划的结构为：

MAX（s , t）= MAX( MAX(s , C[t.row-1][ i ] ) + MAX(C[t.row-1][ i ] , t ) )

点s到t的最大距离为依赖于其子问题MAX(s , C[t.row-1][ i ])，可继续向下划分。

动态规划的几点思考：

1. 动态规划是将无脑的递归计算转换为了循环计算，保存了重复子问题的解，当然也可以写成递归的形式，但看起来不如循环直截了当。最终降低了渐进运行时间。
2. 书中提到动态规划是使用了更多的空间来降低时间复杂度，那么具体是使用了多少更多的空间呢？

对于暴力递归算法，其空间使用主要在与函数在栈上的展开，每次展开均为子问题（n-1）时，使用的栈空间最大，最大时为O（n）,其余子问题均在该空间以内不断压栈和出栈，但是该过程的时间复杂度为指数次，因此相对于动态规划的空间复杂度O（n^2）时间复杂度为多项式次，还是相当划算的。

1. 当面对一个问题时，如何判断是否该用动态规划？
   1. 问题错综复杂，互相包含，无法穷其所有情况（情况太多，人脑难以考虑清楚），暴力递归时，有指数个子问题。
   2. 平行的子问题之间互不影响（无关），而子问题与父问题之间存在包含（递归），并且会对子问题重复求解（重叠）。

4. 如何使用动态规划求解？

最重要的，在于找到其子问题的结构，即父问题与子问题间的关系式。该过程要避免人脑直接考虑问题的解法，因为人脑自动考虑出的极端情况下的解法会被其他情况推翻，因此直接暴力循环解决所有子问题最好不过。