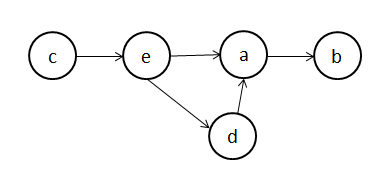
答：任意n个结点的有向无环图都可以得到一个拓扑序列。设拓扑序列为v0v1v2…vn-1，我们来证明此时的邻接矩阵A为上三角矩阵。证明采用反证法。   
假设此时的邻接矩阵不是上三角矩阵，那么，存在下标i和j（i>j），使得A[i][j]不等于零，即图中存在从vi到vj的一条有向边。由拓扑序列的定义可知，在任意拓扑序列中，vi的位置一定在vj之前，而在上述拓扑序列v0v1v2…vn-1中，由于i>j，即vi的位置在vj之后，导致矛盾。因此命题正确

答：解题的主要思路就是推断病毒发作后文档D中的字母顺序，与字典序的字母顺序进行一一对应即可。

具体方法就是从相邻的两个字符串之间推断字母之间的顺序，如题目中单词cac在单词ecd之前，所以可以推断被替换后的字母顺序中字母c在字母e之前，将字母之间的顺序关系看成有向边，画出字母关系图，然后对图进行拓扑排序。得出的序列就是病毒发作后文档D中的字母顺序。

该题中可求得字母的顺序关系图如下：

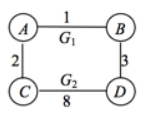


通过拓扑排序，得到的病毒发作后文档D中的字母顺序{c，e，d，a，b}，而字典序为{a，b，c，d，e}，所以可以判断病毒将字母a替换为字母c，将字母b替换为字母e，将字母c替换为字母d，将字母d替换为字母a，将字母e替换为字母b

最终文档D的原内容如下：{abeceda，ada，bac，cad，ded，eda}

答：不正确。

考虑以下含4个结点的图，若划分为V1={A, B}, E1={(A,B)}, V2={C, D}, E2={(C,D)}，最终生成树中的边为{(A,B), (C,D), (A,C)}，而最小生成树中的边为{(A,B), (A,C), (B,D)}。



答：a) 不可行。

能举例说明由于加上常数后，结果出错即可。

比如：

p1(v1,v2,3),(v2,v3,4)

p2(v1,v3,8)

都是从v1到v3路径。最短路径应该选p1。若此时整个图中存在某个负权为-5，则均加上5，那么再利用算法时，选择了p2(v1,v3,13)，而p1(v1,v2,8)(v2,v3,9)不选。

b) 可行。

因为当一个点v在已知集合S中，则在以后的算法步数中，不再检查经过点v的路径，但是因为存在负权边，可能v的权重会变的更小，此时所有经过v到达的点，权值均可能改变。则按照改进的算法，v会重新加入T，会重新计算，则最后肯定能得到正确的结果。

复杂度：题中已经说明存储结构是邻接表，使用最小堆存储源点到各点的路径。则Dijkstra算法复杂度是O( (m+n)logn )(m表示边数，n表示点数)。现在存在重复计算，对于每个新加入S的节点，最坏情况是已加入S的节点全部需要更新，记f(n)表示新加入节点n的更新次数，f(1)=1, f(n)=1+f(1)+f(2)+…+f(n-1)。f(1)+f(2)+…+f(n)=(2^n)-1。

构造s到i权值为i，j到i满足j>i权值为i-2^j可达到指数复杂度。时间复杂度是稠密不用堆O(2^n\*n)、稠密用堆O(2^n\*n\*logn)、稀疏用堆O(2^n\*m/n\*logn)。

答：原式可转化为判断是否存在-log(A[i1,i2])-log(A[i2,i3])-…-log(A[ik,i1]) < 0，即判断图A中是否存在负权环，可使用Bellman-Ford算法或Floyd算法解决，时间复杂度为O(n3)。