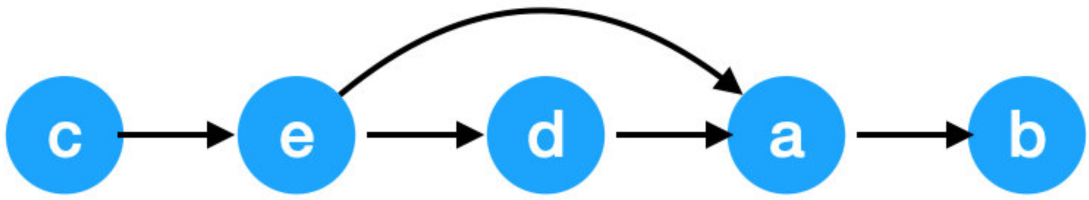
**1.**

对于任意有向无环图，其结点均可排列成一个拓扑序列，设其为$v_0 v_1 v_2 \ldots v_n$，则该序列对应的邻接矩阵主对角线以下的元素全部为0。

否则，若存在$w[i][j] \neq 0 \land i > j$，则存在$v_i$到$v_j$的边，但由于$v_0 v_1 v_2 \ldots v_n$是拓扑序列，所以$v_i$到$v_j$无边，产生矛盾。

**2.**

由病毒发作后文档D中的内容知，病毒发作后的字母顺序关系图为：



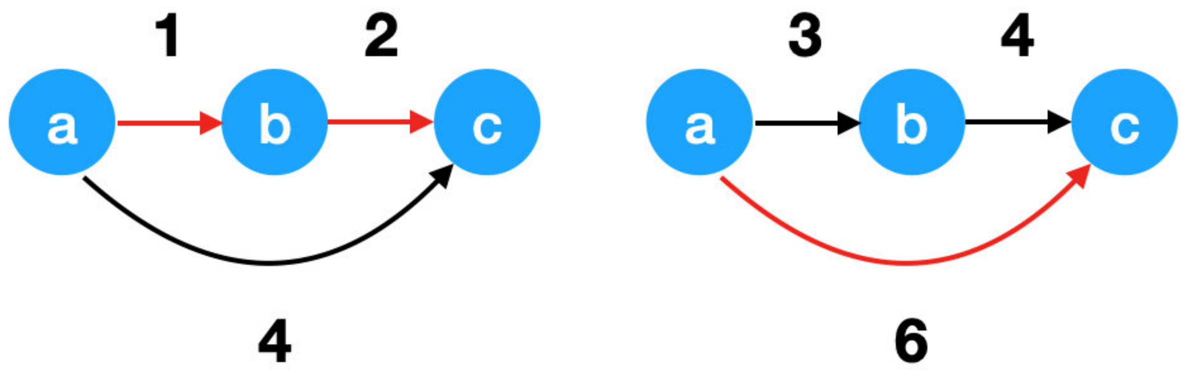

对该图进行拓扑排序，得{c, e, d, a, b}，此即为病毒发作后的字母集。因此病毒发作时的字母替换规则为：a-c, b-e, c-d, d-a, e-b。因此原文档为{abeceda, ada, bac, cad, ded, eda}

**3.**

证明：由题意知，递归出口为$|V_1| = 1, |V_2| = 2$或$|V_1| = |V_2| = 1$或$|V_1| = 2, |V_2| = 1$,此时由$V_1, V_2$显然可以构造出最小生成树。在一般情况下，由MST的性质知，$V_1, V_2$构成的最小生成树必定包含$V_1, V_2$之间权值最小的边，因此每一步所添加的边都是构成完整的最小生成树所必须的，因此算法正确。

**4.**

a). 不可行。




如左图所示，此时a到c的最短路径为a->b->c。若图中其他位置存在一个权值为-2的边，导致所有边的权值均增加2，则如右图所示，a到c的最短路径变成a->c。

b). 可行。

对于已知集合S中的任意结点v，在之后的算法步骤中，由于存在负权值边，v的最短加权路径可能会变小，此时所有经过v到达的结点权值会改变。按照改进的算法，若v的最短加权路径变小，则将v从S中剔除并重新计算，避免了上述问题。

在最坏的情况下，加入结点j到S中时，S中所有的结点i (i<j)全部都要被剔除再重新加入。更新j之前所有结点的用时有如下递推关系式：$t_j = \sum t_i, 0 < i < j, t_1 = 1$。可以得到$t_n = 2^{n-1}$。更新结点j自己的距离时，时间复杂度为$\mathcal{O}(n \log n)$，删除前面所有节点的时间复杂度为$\mathcal{O}(n)$，重新更新j前面所有结点的时间复杂度为$\mathcal O(n2^n)$，因此总时间复杂度为$\mathcal{O}(n2^n)$

**5.**

使用Floyd算法的改造版，时间复杂度为$\mathcal O(n^3)$

```
double A[MAX][MAX];
bool Floyd(int n) {
    for (int k = 0; k < n; k++) {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (A[i][j] < A[i][k] * A[k][j])
                    A[i][j] = A[i][k] * A[k][j];
            }
        }
    }
    for (int i = 0; i < n; i++) {
        if (A[i][i] > 1)
            return true;
    }
    return false;
}

```