1.

这个有向图导出边的序列 (a,b,w) (a 是入点, b 是出点, w 是权重). 以及数组 m , 其中 $m[v_i]$ 表示点 v_i 的入度
算法设计如下:

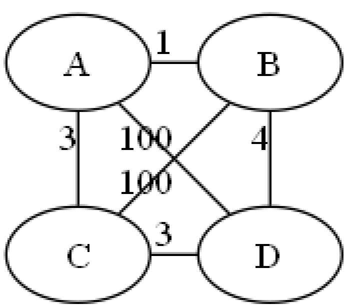
- 建立栈 s , 将 m 值为 0 的点入栈, 建立数组 t 表示完成该点对应项目的最短时间, 初始化为 0
- 若栈不空:
  - 栈顶元素出栈, 对以该元素为入点的所有边 (a,b_i,w_i)
    - $t[b_i]=\max\{t[b_i],t[a]+w_i\}$
    - $m[b_i]$ 减去1, 若操作后为0, 则 b_i 入栈
- 栈清空后数组 t 的元素的最大值即为在满足所有的依赖条件的情况下完成所有项目的最短时间

2.

(1)
是的, Dijkstra单源最短路径算法每增加一个节点即连一条边, 直到全部顶点与源联通, 这表面生成结果覆盖全部顶点且连通, 且边数 = 节点数 - 1
由于覆盖且连通全部顶点, 并且边数 = 节点数 - 1, 因此该生成结果无环, 证毕

(2)

考虑在如下图中以A为中心运行Dijkstra单源最短路径算法:



得到 $A\to C, A\to B\to D$, 这是一棵生成树且包含了 A 到所有点的最短路径, 但与唯一最小生成树 $D-C-A-B$ 所连的边不符 (实际上路径权重和也不符).

3.

(1)
$(A,B),(B,E),(E,F),(D,E),(C,E)$

(2)
$(E,F),(D,E),(B,E),(A,B),(C,E)$

(3)

如果不唯一, 假设 T_1 和 T_2 都是最小生成树. 假设 $e = \underset{e_i \in T_1 \cup T_2, e_i \notin T_1 \cap T_2}{\arg\min} w(e_i)$.

不妨设 $e \in T_1, e \notin T_2$, 将 e 添加到 T_2 则后者成环, 注意到这个环不可能在 T_1 中完整出现, 因此存在 $e' \in T_2 \setminus T_1$, e' 在 e 引起的这个环上.

此时, $T_1 \cup \{e'\} \setminus \{e\}, T_2 \cup \{e\} \setminus \{e'\}$ 都是生成树, 注意到 T_1 和 T_2 是最小生成树, 因此 $w(e') - w(e) \geq 0, w(e) - w(e') \geq 0$, 从而 $w(e) = w(e')$. 这与假设中的边权值互不相同矛盾. 证毕