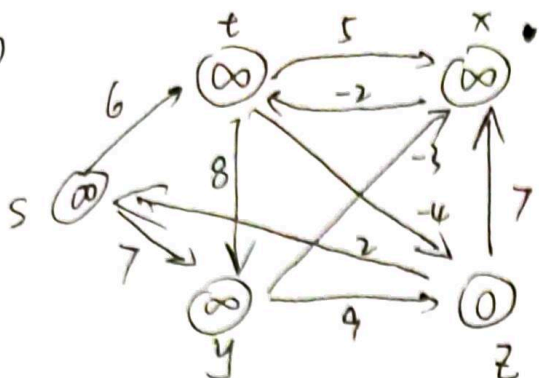
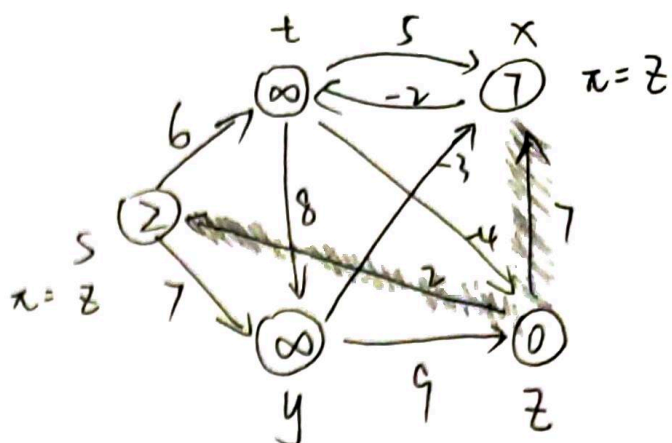


24.1-1

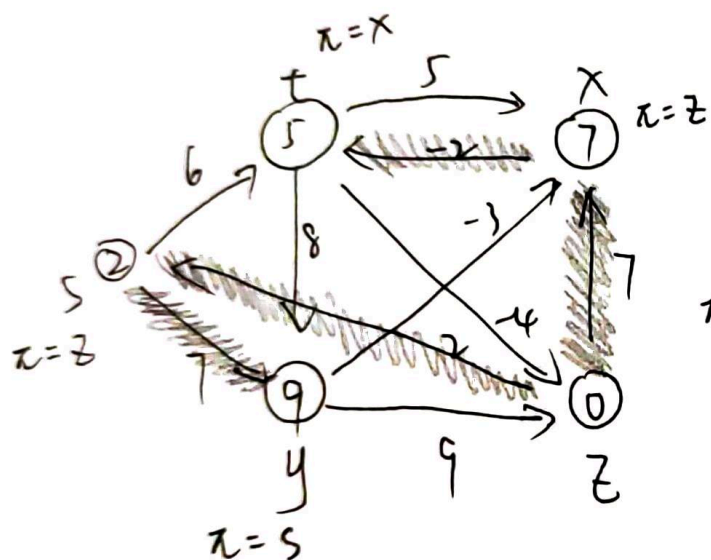
(1)



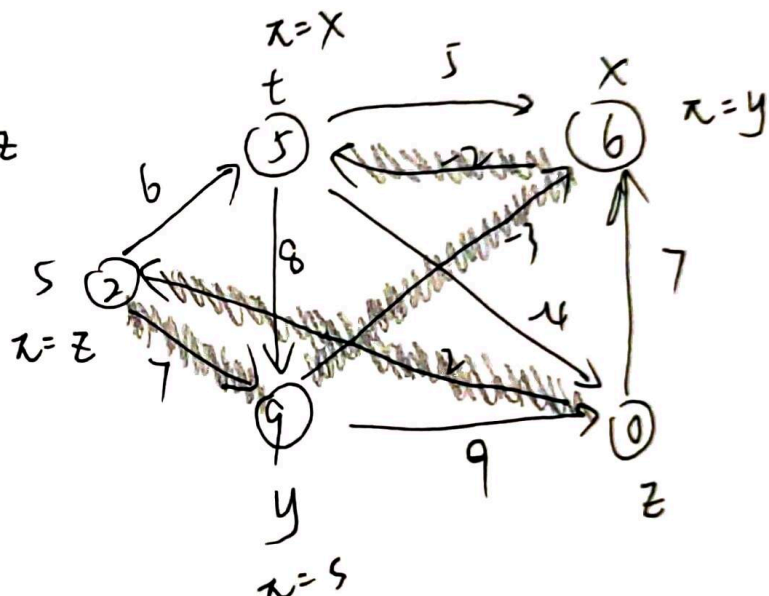
(a)



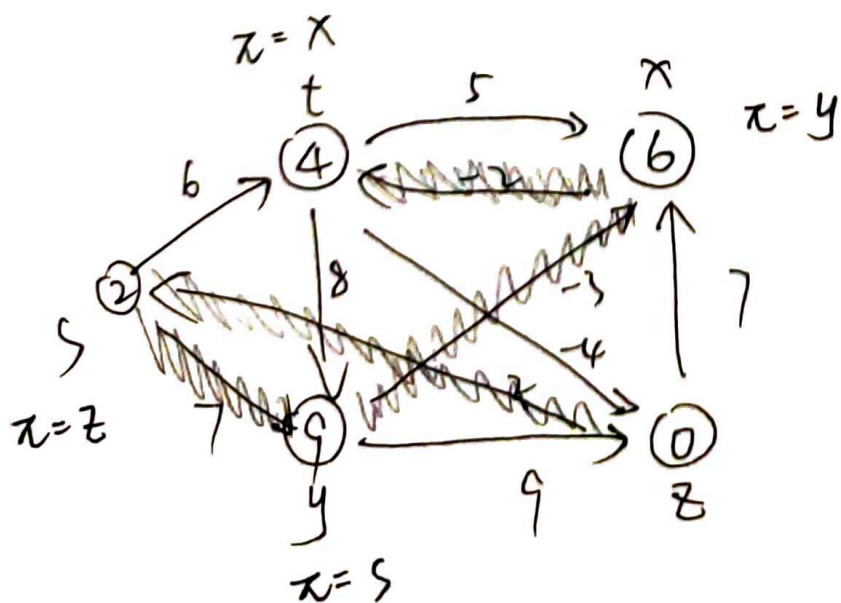
(b)



(c)

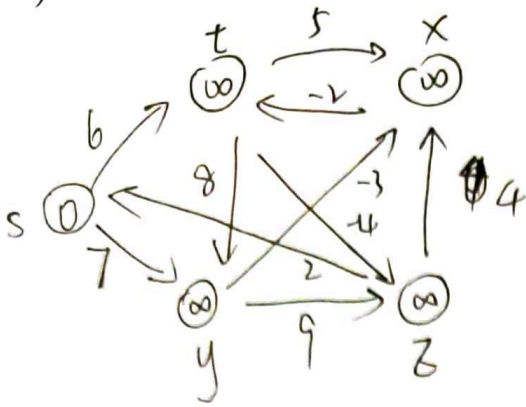


(d)

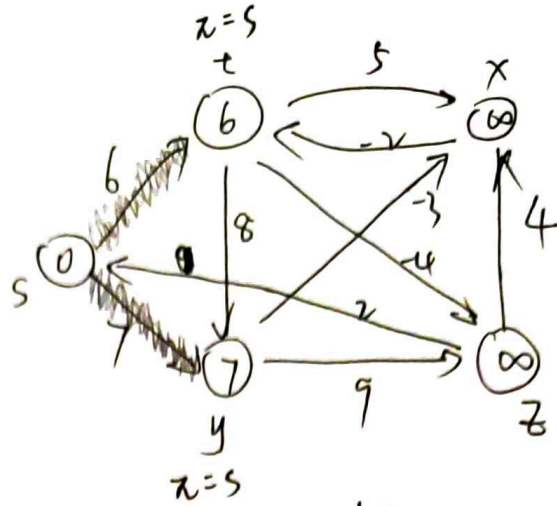


(e)

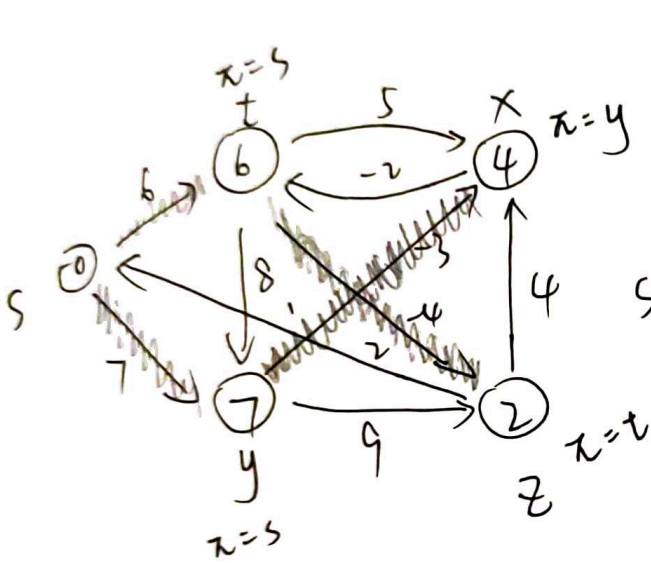
(2)



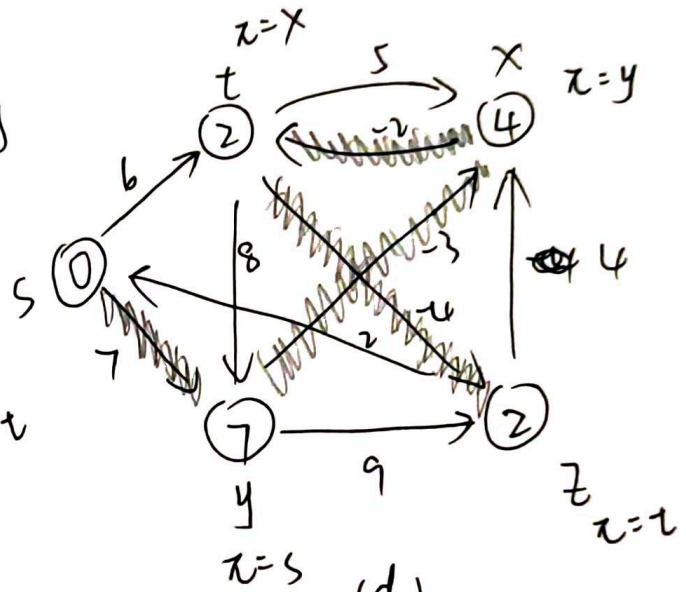
(a)



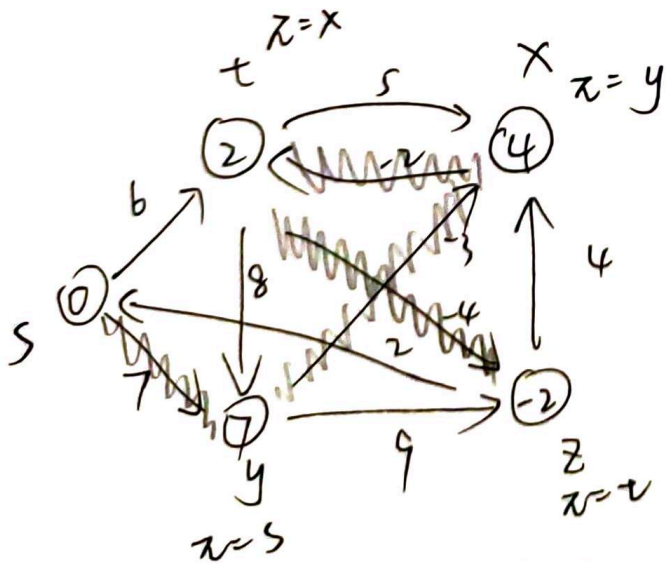
(b)



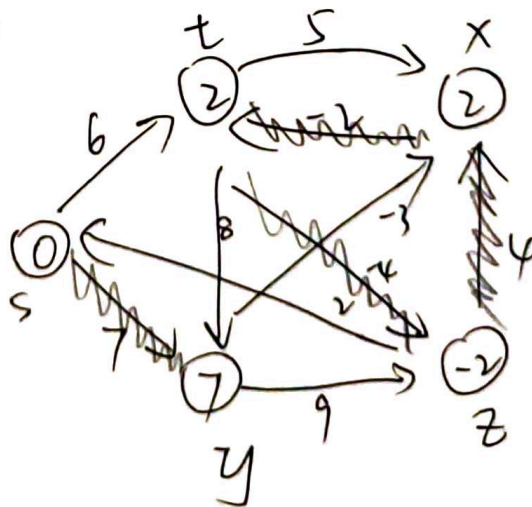
(c)



(d)



(e) 有负环  $(z, x, t)$ .



(f) 构成负环.

24.1-6.

算法：同 Bellman-Ford 算法类似。

- ① 初始化
- ② 对每条边进行  $|V| - 1$  次松弛操作。
- ③ 再对所有边进行一次松弛。此时如果还能更新某个节点的最短路，则存在负环。
- ④ 从能被更新的那个节点开始利用前驱节点  $v$  进行回溯，找到构成负环的节点。

正确性：

首先从定理 24.4 可知 Bellman-Ford 算法的正确性。

可以正确检测是否有负环。

那么在松弛第  $|V|$  次时，可知负环上的节点一定会被更新。且它的前驱节点为负环上的节点。同理负环上的每一节点在松弛后都能得到更新，所以它们的前驱节点一定构成一个负环。即算法正确。