

Homework 8  
 Yaqiao Liu  
 NUID 001279512

### Question 2

1.

Firstly, I show the initial, intermediate and final state of outer loop.

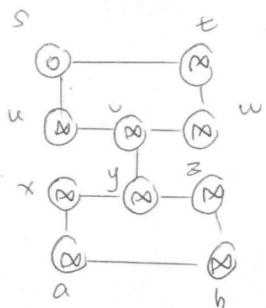
Then I show the inner loop in 1<sup>st</sup> and 2<sup>nd</sup> outer loop iterations, because it would converge in the second cycle.

Red letter below the graph represents the current node. In other word, it is the inner loop position.

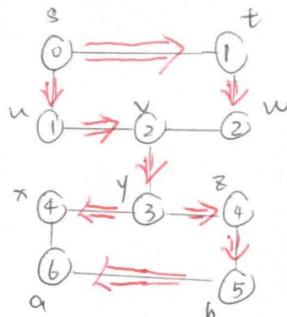
Red arrow represents the parent direction. And the distances are shown in the circle.

I assume all the nodes are stored in memory orderly. (like s,t,u,v,w,x,y,z,a,b) The saving order would influence transverse order. So it is important in this problem.

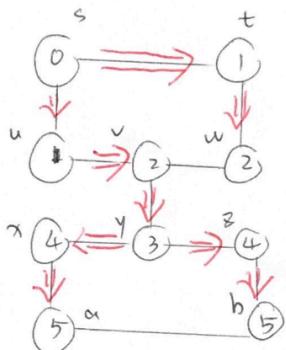
Outer Loop:



initial.



After  
intermediate (1st Outer loop)



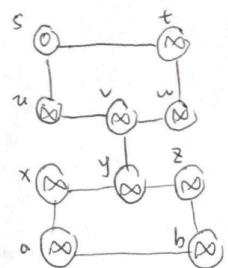
Final . → (After second outer loop)

Inner Loop:

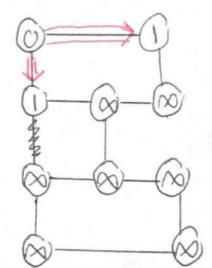
Outer Loop  
Iteration.

I

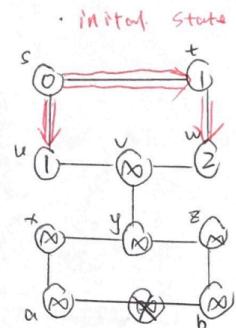
(a)



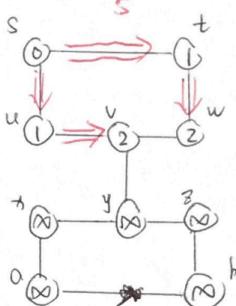
(b)



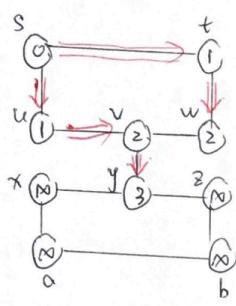
(c)



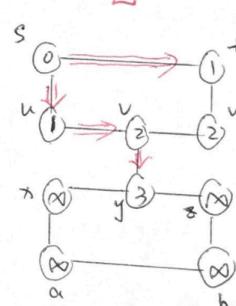
(d)



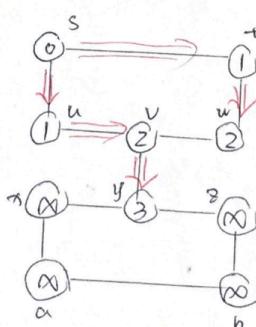
(e)



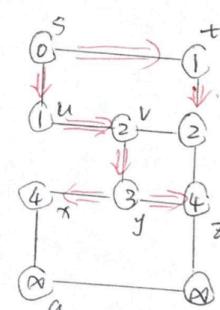
(f)



(g)

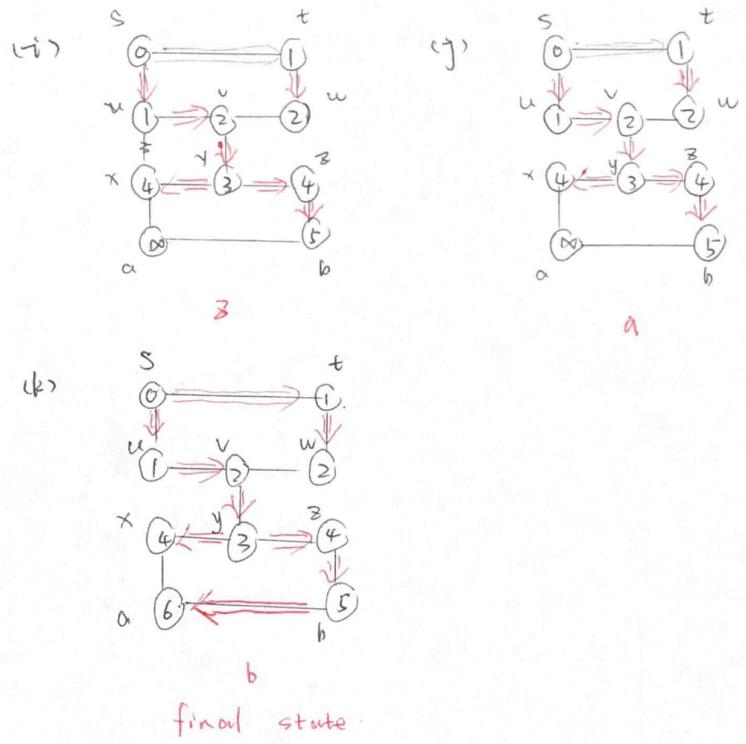


(h)



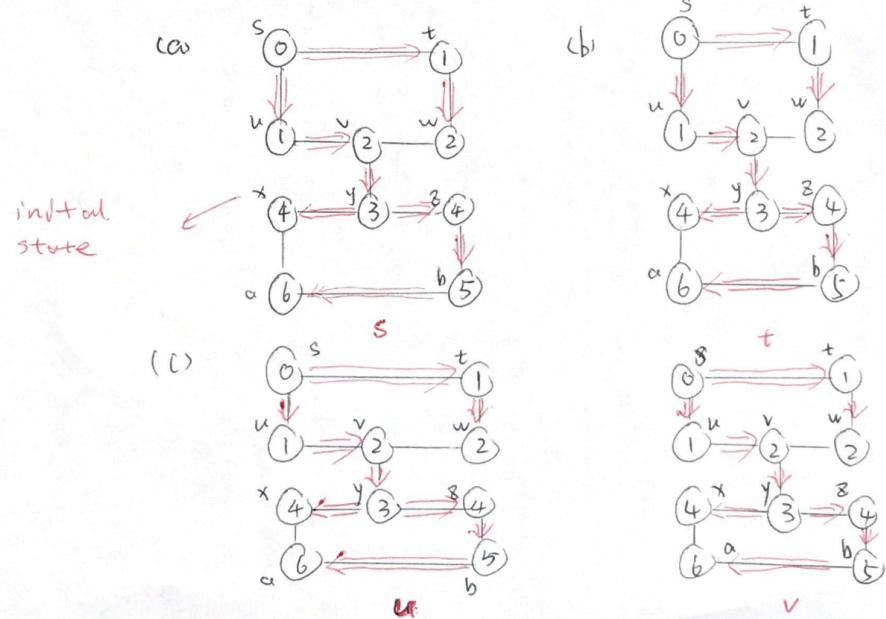
x

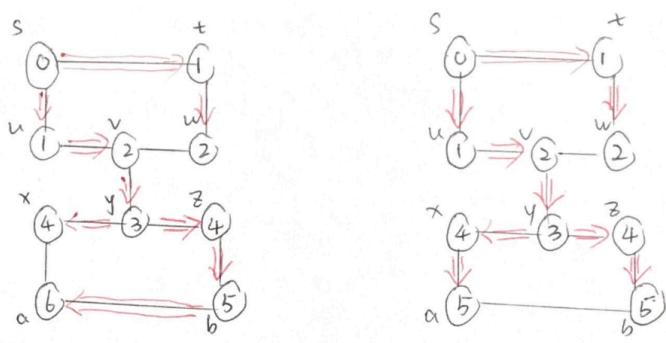
y



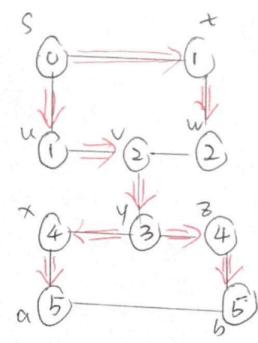
Iteration

II

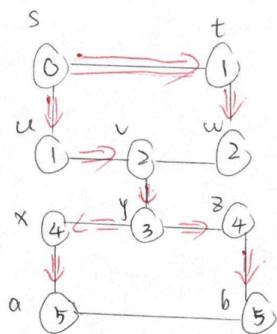




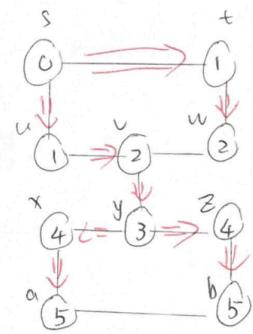
w.



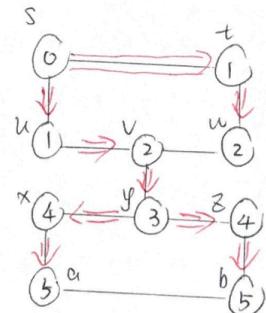
~~w~~



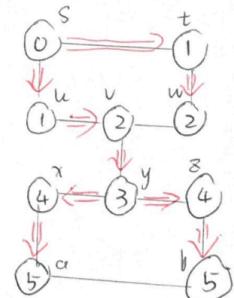
y



~~y~~

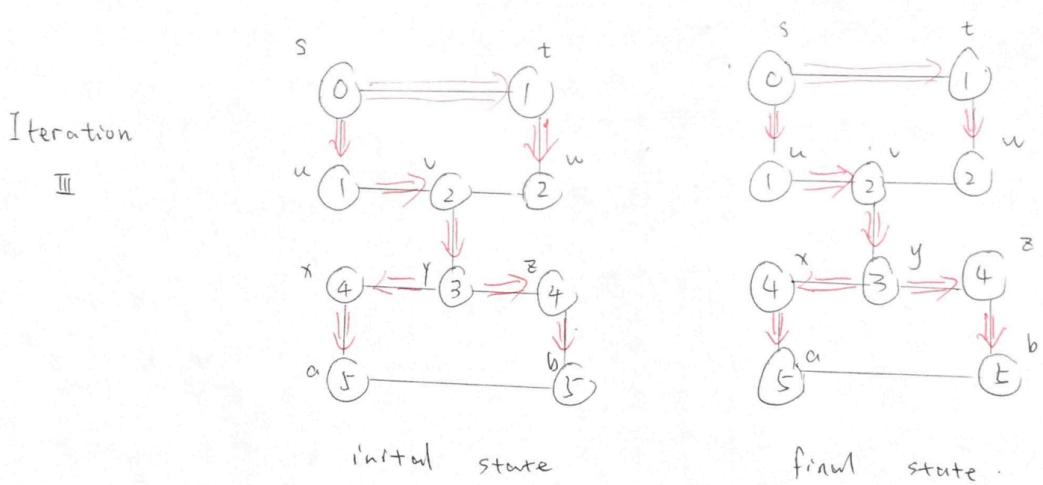


a



b

final state



c.

I have two solutions for lower time cost.

The first one is that I can add an (0/1) variable in Bellman-Ford function to express if the distance of each node will change. If the distance for each node is unchanged, we can stop it after this loop.

The second solution is that we can choose some quicker algorithm. For example, we can choose Dijkstra algorithm to solve this question. The time cost is  $O(|V| + |E|) * \log(|V|)$ . But it has a constrain, this algorithm cannot deal with negative weights.