1)

1) adj[q] = [s, t, w] adj[u] = [y] adj[y] = [q]

adj[r] = [u, y] adj[v] = [w] adj[z] = [x]

adj[s] = [v] adj[w] = [s]

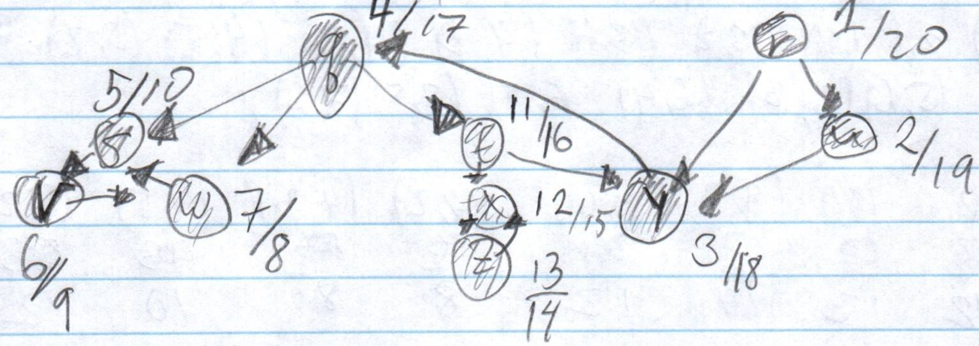
adj[t] = [x, y] adj[x] = [z]

2) BFS $\cancel{s}$ r

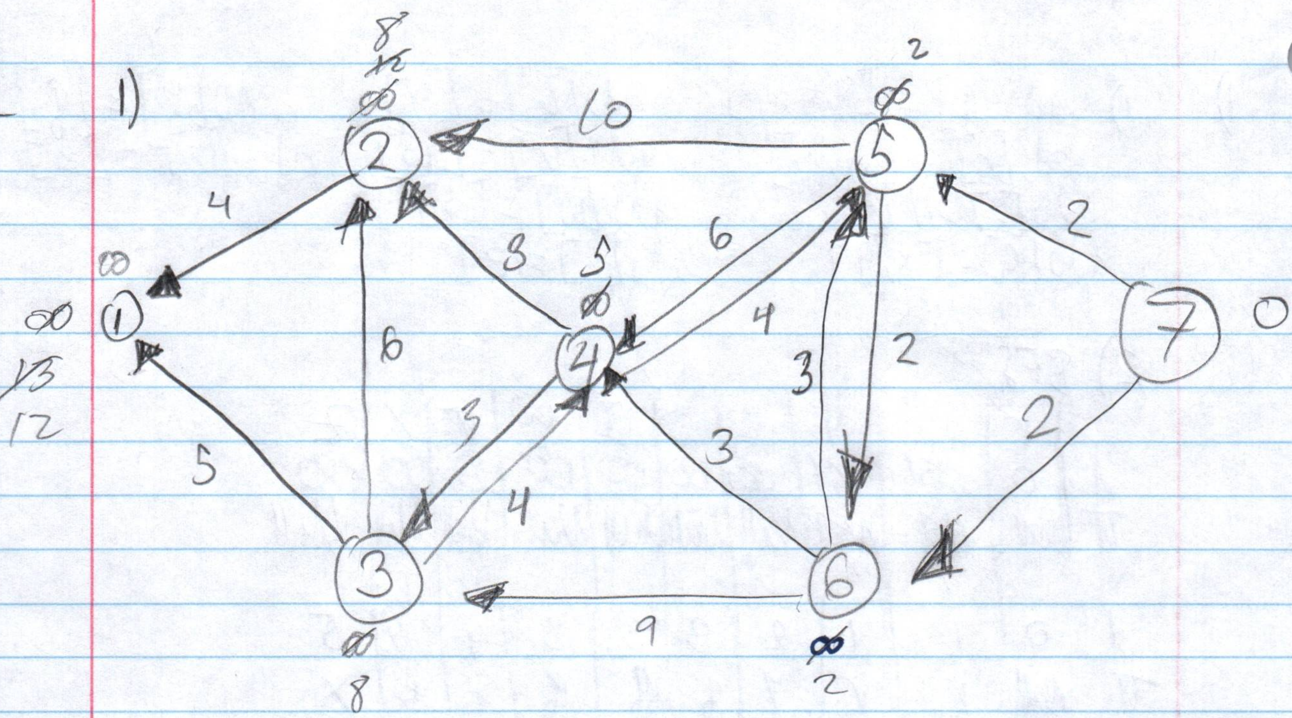
	r	u	y	q	s	t	w	v	x	z
d	0	$\cancel{\infty}$ 1	$\cancel{\infty}$ 1	$\cancel{\infty}$ 2	∞	∞	∞	∞	∞	∞
π	Null	~~Null~~ r	~~Null~~	Null	Null	Null	Null	Null	Null	Null

d	0	1	1	2	3	3	3	4	4	5
π	Null	r	r	y	q	q	q	s	t	x

3)



2 1)



2) (2,1) (3,1) (3,2) (3,4) (4,2) (4,3) (4,5) (5,2) (5,4)
(5,6) (6,3) (6,4) (6,5) (7,5) (7,6)

(2,1)	(3,1)	(3,2)	(3,4)	(4,2)	(4,3)	(4,5)	(5,2)
$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$
12	13	14	12	8	8	10	12

(5,4)	(5,6)	(6,3)	(6,4)	(6,5)	(7,5)	(7,6)
$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$	$\cancel{\infty}$
8	4	11	5	5	2	2

3) Bellman Ford's algorithm as long as there are no negative cycles

$O(VE)$

Every edge is negative weight

$G[V, E]$

S = Source vertex

$d[S] = 0$

d = distance

Relaxation

for E in G[V, E]

if $d[v] < d[u]$

Relax all edges k times

relax (u, v, w)

if $d[v] > d[u] + w(u, v)$

return false

— Negative cycle

4) $G = (V, E, w)$ $s \in V$

$\pi[s] = 0$ to be visited again it must be a cyclic graph, & to be updated for $\pi[s] = 0$ the edge must be negative $\therefore$ it is a negative cycle through s

2) Stop when find vertex s or the vertex is grayed & has been seen before. Stop & output if there is a negative cycle through S so that $\pi[s]$ doesnt get updated