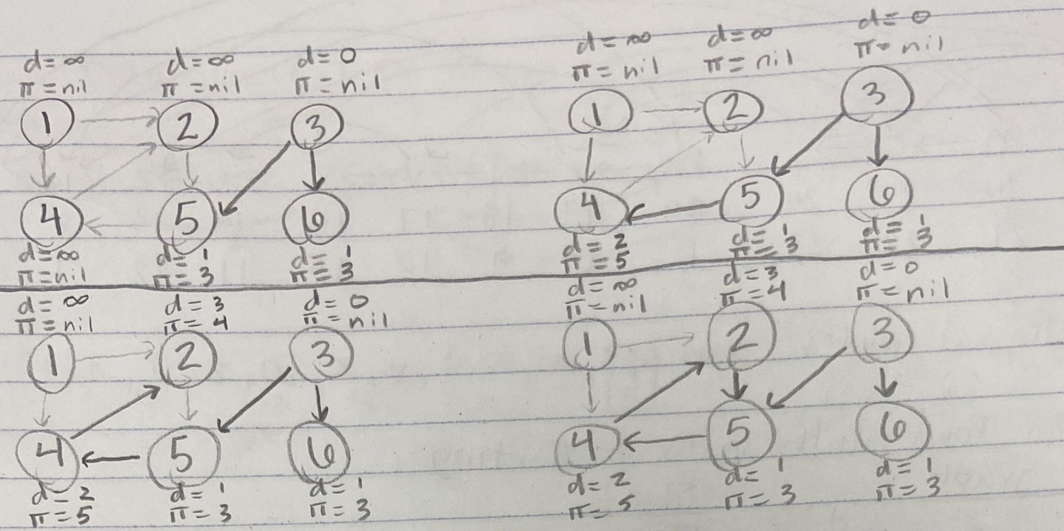
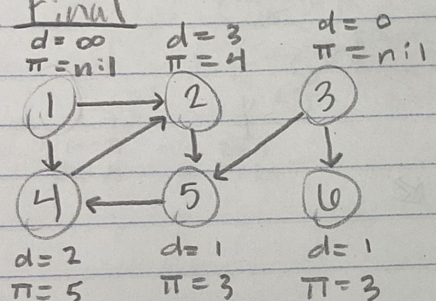
Lauren
Lyons

Homework 4

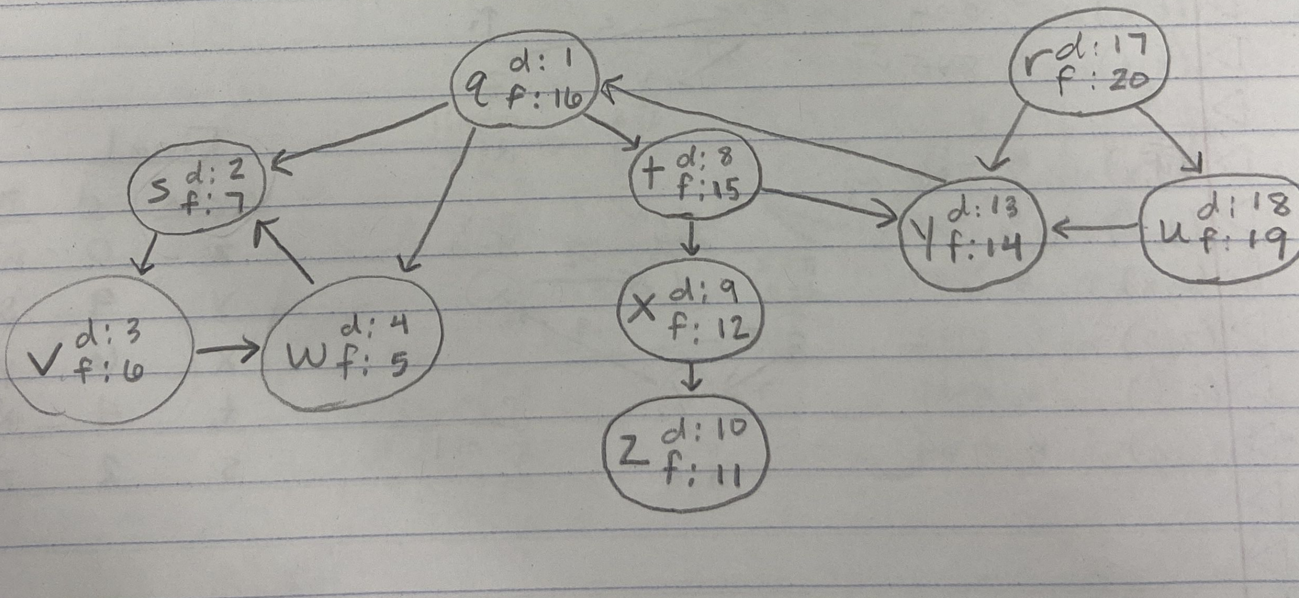
1



Final

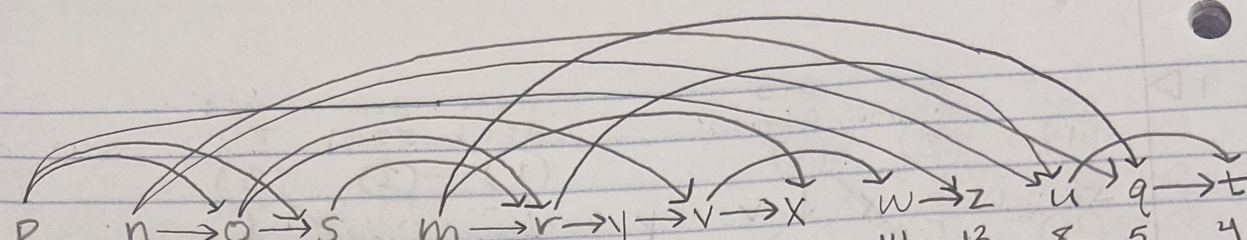


2



Lauren Lyons

3 ▷

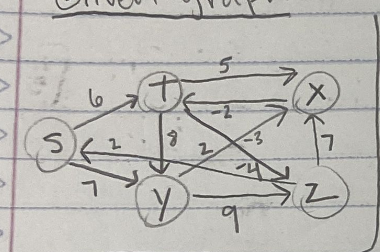


	p	n	o	s	m	r	y	v	x	w	z	u	q	t
f:	28	26	25	24	20	19	18	17	16	14	13	8	5	4
d:	27	21	22	23	1	6	9	10	15	11	12	7	2	3

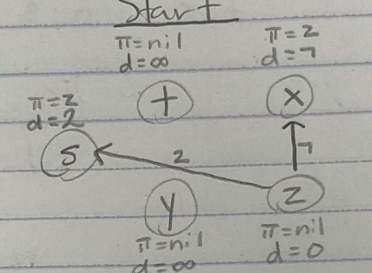
Topological Order: p, n, o, s, m, r, y, v, x, w, z, u, q, t

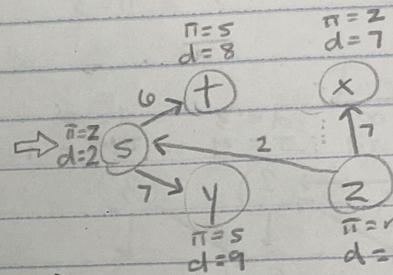
4 ▷ Bellman-Ford with vertex z starting

Given graph



Start





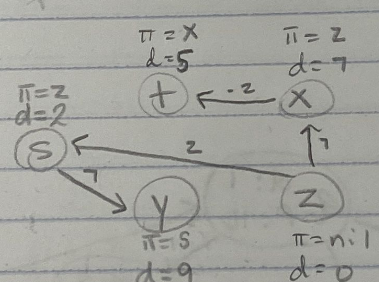
relax

(t,x) no b/c 7→13

(t,y) no b/c 9→16

(t,z) no b/c 0→4

(x,t) yes b/c 8→5



(y,x) yes b/c 7→6

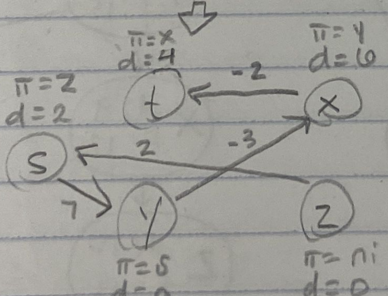
(y,z) no b/c 0→18

(z,x) no b/c 6→7

(z,s) no b/c 2→2

(s,t) no b/c 5→8

(s,y) no b/c 9→9

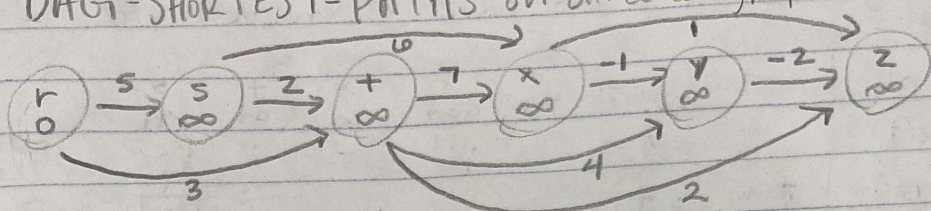


Final

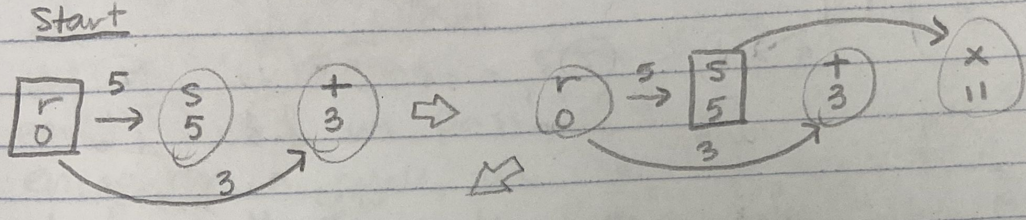
vertex	d	π
z	0	nil
y	9	s
x	6	y
t	4	x
s	2	z

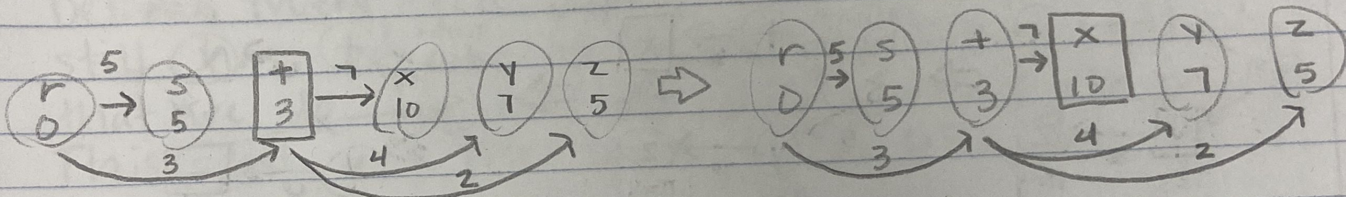
Lauren
Lyons

5 Run DAG-SHORTEST-PATHS on directed graph using vertex r as start



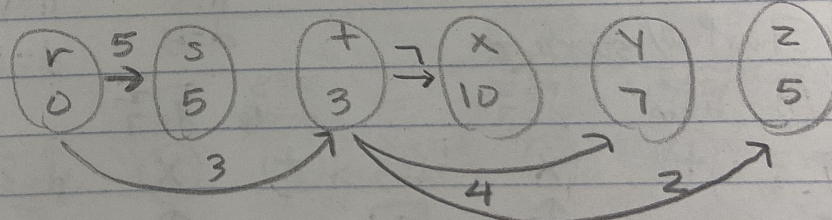
Start





No changes from relaxing for the vertices y and z

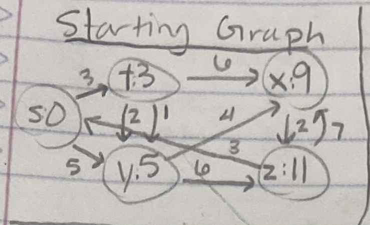
Final



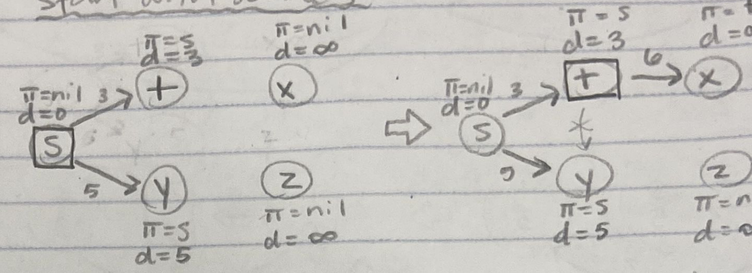
Lauren Lyons

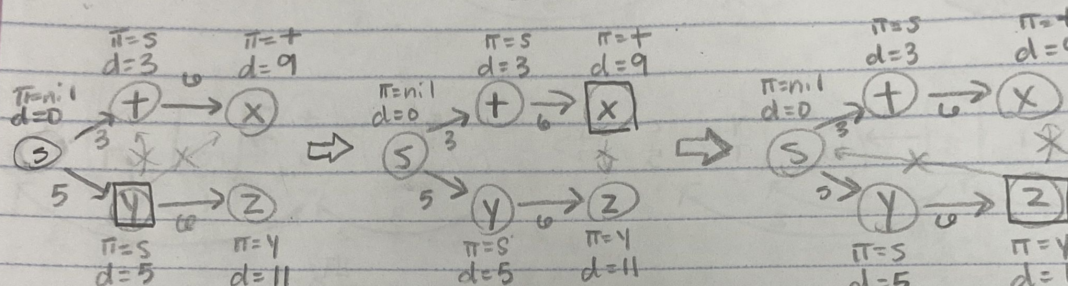
6 ▷ Dijkstra's using vertex s as source then vertex z

Starting Graph

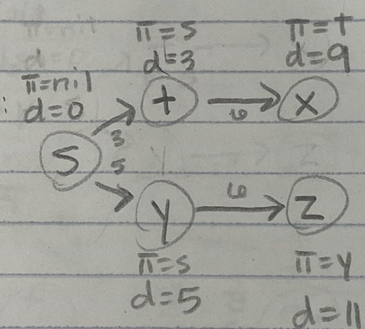


Start with source s

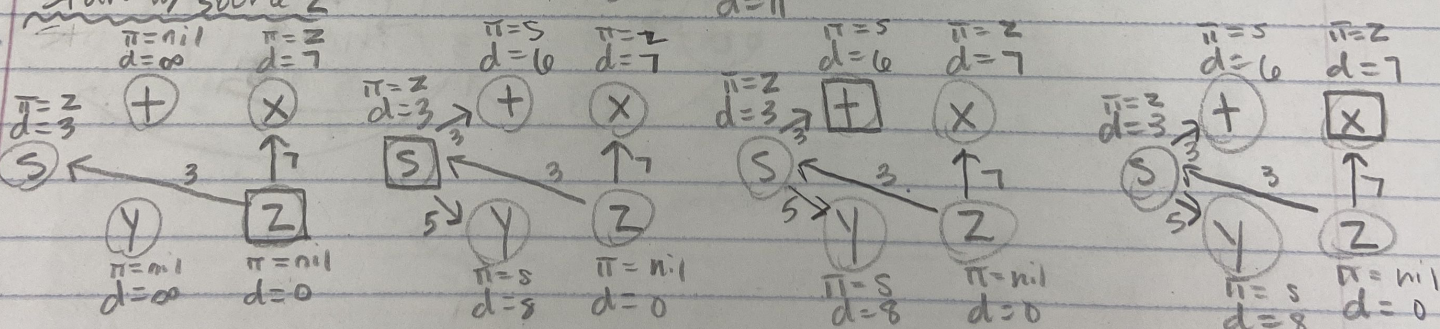


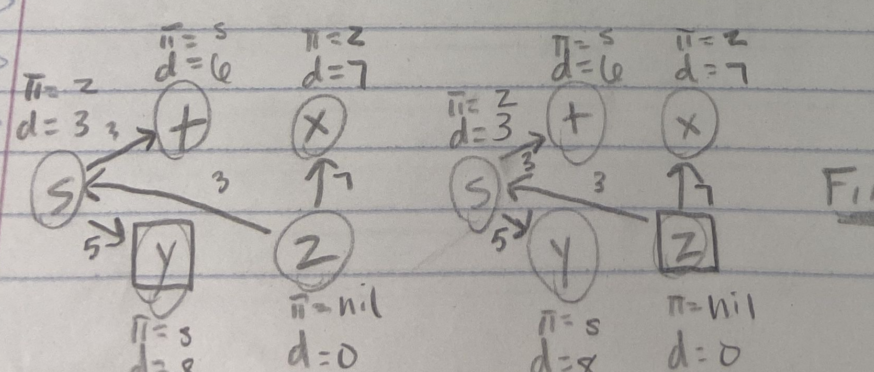


Final:

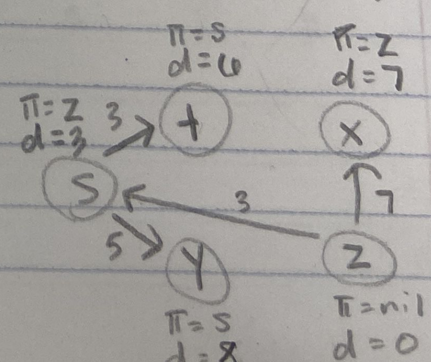


Start w/ source z





Final:



Lauren Lyons

After the new vertex and edges are added to the graph G, it may not be a MST. So, using the DFS, we will check for cycles. The DFS algorithm uses white, gray, and black to indicate what has been visited. Using that, if we visit any vertex that is already gray, that is indicative of a cycle. So, when we are performing a DFS, we need a max-heap for each vertex we pass in the stack trace. So if we encounter a cycle, we can look at any vertex before that cycle detection up until a vertex still has adjacencies to check. That will ensure that we do not cut off any part of the graph. This will cost the same as a DFS, which is $O(|V| + |E|)$.