James Villemarette

CISC 220-012 Dec. 3, 2019

Try-on-Paper Problems

	H	A	U	M	P	F	E	S
H	0	4	3	∞	∞	∞	∞	∞
A	∞	0	∞	2	7	∞	∞	∞
U	∞	6	0	1	6	∞	∞	∞
M	∞	∞	∞	0	2	5	5	∞
P	∞	∞	∞	∞	0	5	4	∞
F	∞	∞	∞	∞	∞	0	∞	2
E	∞	∞	∞	∞	∞	5	0	3
S	∞	∞	∞	∞	∞	∞	∞	0

	H	A	(U)	M	P	F	E	S
Dist	0	4	3	∞	∞	∞	∞	∞
Prev	H	H	H	H	H	H	H	H
Visited?	✓	x	x	x	x	x	x	x

↓

	H	A	U	(M)	P	F	E	S
Dist	0	4	3	4	9	∞	∞	∞
Prev	H	H	H	U	U	H	H	H
Visited?	✓	x	✓	x	x	x	x	x

↓

	H	(A)	U	M	P	F	E	S
D	0	4	3	4	6	9	9	∞
P	H	H	H	U	M	M	M	H
V	✓	x	✓	✓	x	x	x	x

↓

					(P)			
D	0	4	3	4	6	9	9	∞
P	H	H	H	U	M	M	M	H
V	✓	✓	✓	✓	x	x	x	✓

→

2

	H	A	U	m	F	E	S
D	0	4	3	4	6	9	9	∞
P	H	H	H	U	m	m	m	H
V	✓	✓	✓	✓	✗	✗	✗	✗

↓

D	0	4	3	4	6	9	9	∞
P	H	H	H	U	m	m	m	H
V	✓	✓	✓	✓	✓	✗	✗	✗

↓

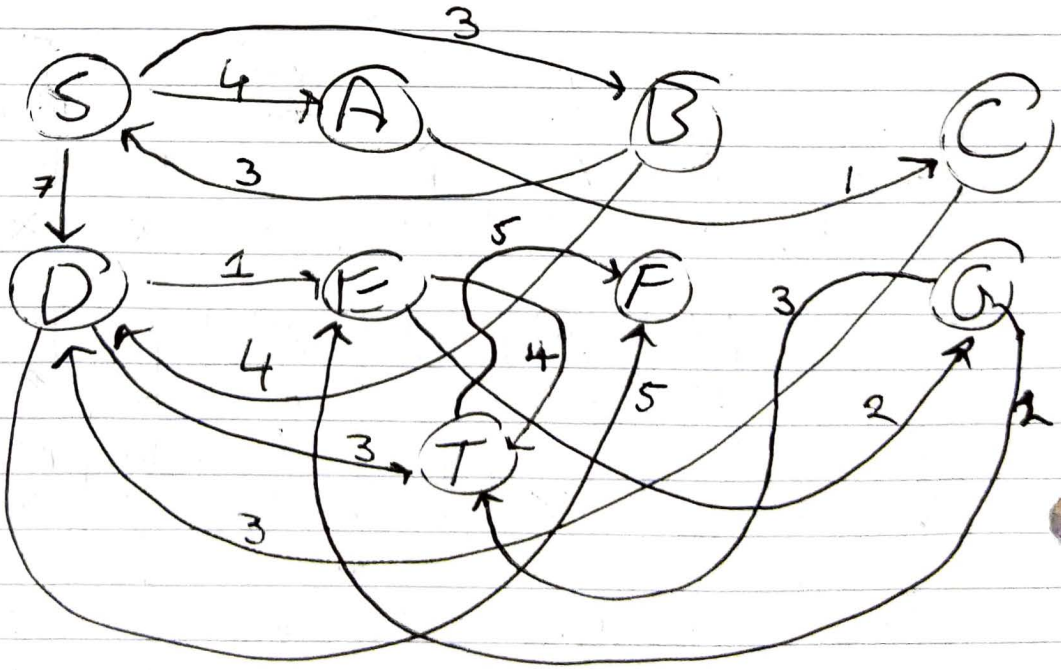
D	0	4	3	4	6	9	9	12
P	H	H	H	U	m	m	m	E
V	✓	✓	✓	✓	✓	✗	✓	✗

↓

D	0	4	3	4	6	7	7	11
P	H	H	H	U	m	m	m	F
V	✓	✓	✓	✓	✓	✓	✓	

H → U → m → F

2)



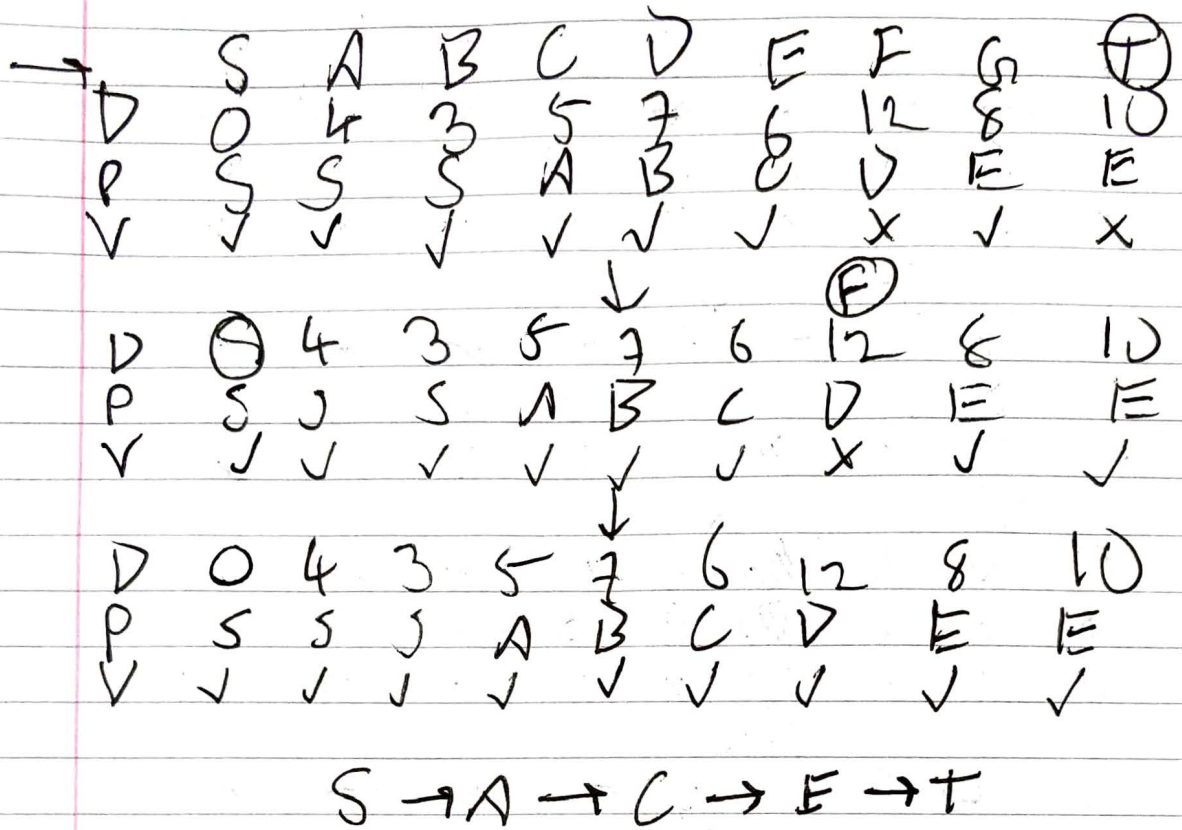
James Villemaire

CISC 220-012 Dec. 3, 2019

3)

	S	A	B	C	D	E	F	G	T
D	0	∞	∞	∞	∞	∞	∞	∞	∞
P	S	S	S	S	S	S	S	S	S
V	✓	✗	✗	✗	✗	✗	✗	✗	✗
			(B) ↓						
D	0	4	3	∞	7	∞	∞	∞	∞
P	S	S	S	S	S	S	S	S	S
V	✓	✗	✗	✗	✗	✗	✗	✗	✗
		(A)		↓					
D	0	4	3	~~∞~~ 7	7	∞	∞	∞	∞
P	S	S	S	~~S~~ B	B	S	S	S	S
V	✓	✗	✓	~~✗~~ ✗	✗	✗	✗	✗	✗
				(C) ↓					
D	0	4	3	5	7	∞	∞	∞	∞
P	S	S	S	A	B	S	S	S	S
V	✓	✓	✓	✗	✗	✗	✗	✗	✗
					↓	(E)			
D	0	4	3	5	7	6	∞	∞	∞
P	S	S	S	A	B	C	S	S	S
V	✓	✓	✓	✓	✗	✗	✗	✗	✗
					↓	(D)			
D	0	4	3	5	7	6	∞	8	10
P	S	S	S	A	B	C	S	E	E
V	✓	✓	✓	✓	✗	✓	✗	✗	✗
				↓				(G)	
D	0	4	3	5	7	6	12	8	10
P	S	S	S	A	B	C	D	E	E
V	✓	✓	✓	✓	✓	✓	✗	✗	✗
				↓					(T)
D	0	4	3	5	7	6	12	8	10
P	S	S	S	A	B	C	D	E	E
V	✓	✓	✓	✓	✓	✓	✗	✓	✗ →

4



4)

	S	A	B	C	D	E	F	G	T
S	0	4	3		7				
A	4	0							
B	3		0		4				
C				0	3	1			
D	7		4	3	0	1	5		3
E				1	1	0		2	4
F					5		0		5
G						2		0	3
T					3	4	5	3	0

5) B/c Dijkstra's algo. would keep pointing back to that vertex w/ the negative value in many circumstances. It also doesn't work logically: it'd be like a toll road that paid you to drive on it. Dijkstra assumes to visit each vertex once, whereas a negative vertex would loop through that vertex to decrease overall cost.

James Villemaire

CISC 220-012 Dec. 5, 2014

6) B/C the largest-cost path would be infinitely big where it loops through the nodes repeatedly in an adjacency matrix of three or more vertices: