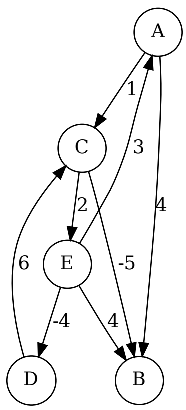
CS5200 Homework 4 Graphs
Adam McNeil
Question 1



D(0) =

	A	B	C	D	E
A	0	4	1	∞	∞
B	∞	0	∞	∞	∞
C	∞	-5	0	∞	2
D	∞	∞	6	0	∞
E	3	4	∞	-4	0

$\pi(0)$ =

	A	B	C	D	E
A	nil	A	A	nil	nil
B	nil	nil	nil	nil	nil
C	nil	C	nil	nil	C
D	nil	nil	D	nil	nil
E	E	E	nil	E	nil

D(1) =

	A	B	C	D	E
A	0	4	1	∞	∞
B	∞	0	∞	∞	∞
C	∞	-5	0	∞	2
D	∞	∞	6	0	∞
E	3	4	4	-4	0

$\pi(1)$ =

	A	B	C	D	E
A	nil	A	A	nil	nil
B	nil	nil	nil	nil	nil
C	nil	C	nil	nil	C
D	nil	nil	D	nil	nil
E	E	E	A	E	nil

D(2) =

	A	B	C	D	E
A	0	4	1	∞	∞
B	∞	0	∞	∞	∞
C	∞	-5	0	∞	2
D	∞	∞	6	0	∞
E	3	4	4	-4	0

$\pi(2)$ =

	A	B	C	D	E
A	nil	A	A	nil	nil
B	nil	nil	nil	nil	nil
C	nil	C	nil	nil	C
D	nil	nil	D	nil	nil
E	E	E	A	E	nil

D(3) =

	A	B	C	D	E
A	0	-4	1	∞	3
B	∞	0	∞	∞	∞
C	∞	-5	0	∞	2
D	∞	1	6	0	8
E	3	-1	4	-4	0

$\pi(3)$ =

	A	B	C	D	E
A	nil	C	A	nil	C
B	nil	nil	nil	nil	nil
C	nil	C	nil	nil	C
D	nil	C	D	nil	C
E	E	C	A	E	nil

D(4) =

	A	B	C	D	E
A	0	-4	1	∞	3
B	∞	0	∞	∞	∞
C	∞	-5	0	∞	2
D	∞	1	6	0	8
E	3	-3	2	-4	0

$\pi(4) =$

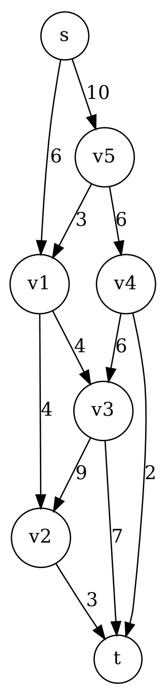
	A	B	C	D	E
A	nil	C	A	nil	C
B	nil	nil	nil	nil	nil
C	nil	C	nil	nil	C
D	nil	C	D	nil	C
E	E	D	D	E	nil

D(5) =

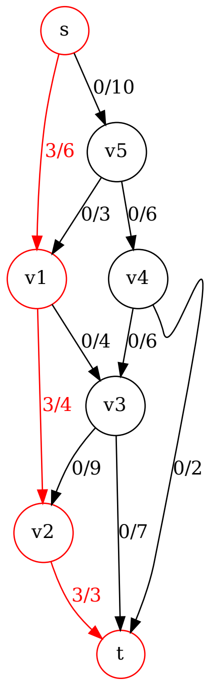
	A	B	C	D	E
A	0	-4	1	-1	3
B	∞	0	∞	∞	∞
C	5	-5	0	-2	2
D	11	1	6	0	8
E	3	-3	2	-4	0

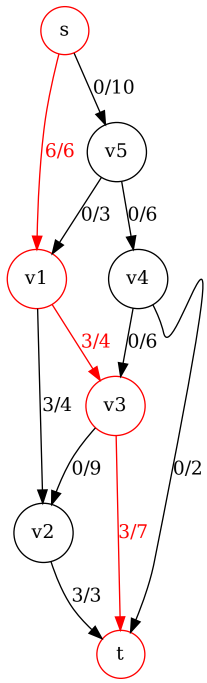
$\pi(5) =$

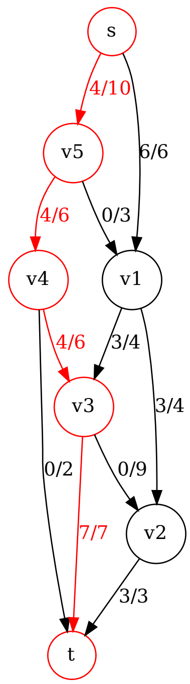
	A	B	C	D	E
A	nil	C	A	E	C
B	nil	nil	nil	nil	nil
C	E	C	nil	E	C
D	E	C	D	nil	C
E	E	D	D	E	nil

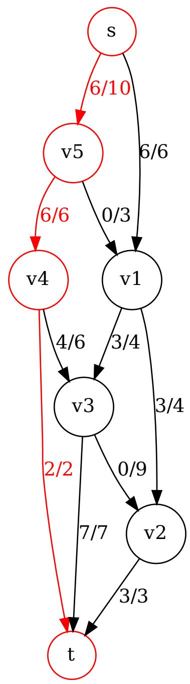


Question 2


s
0/10
3/6
v5
0/3
0/6
v1
v4
0/4
0/6
3/4
v3
0/9
0/2
v2
0/7
3/3
t


s
0/10
6/6
v5
0/3
0/6
v1
v4
3/4
0/6
3/4
v3
0/9
0/2
v2
3/7
3/3
t


s
4/10
6/6
v5
0/3
4/6
v4
v1
4/6
3/4
v3
3/4
0/2
0/9
7/7
v2
3/3
t



Question 3

Strategy A: 5 different paths until the final answer is reached

$s- > v1- > v2- > t$

Question 4

1) For a given cut (S, T), the net flow from S to T can be greater than capacity of S and T.

False

2) For any (S, T) cut, if the net flow equals to the capacity of S and T, then we cannot find any augmenting path in the residual graph.

True

3) The Floyd-Warshall algorithm belongs to the greedy algorithm, as it is more efficient than the dynamic programming solution.

False

4) The Dijksta's algorithm can be used to find the all-pairs shortest paths in a weighted directed graph, and it is more efficient than some dynamic programming solution.

True