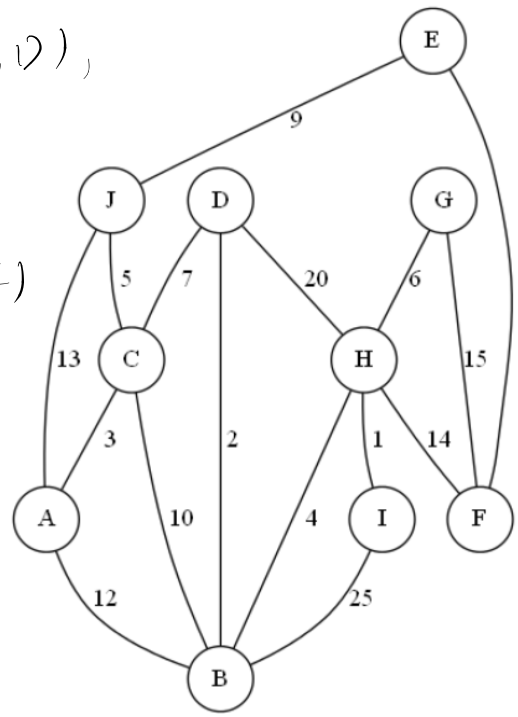
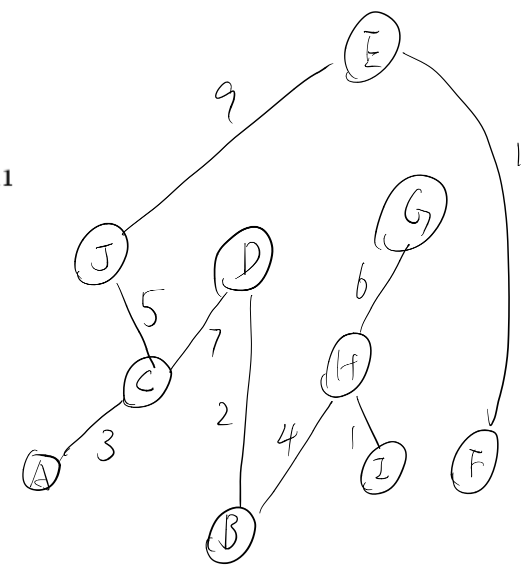
INT202 Complexity of Algorithms

1. Use Prim's algorithm starting at node A to compute the Minimum Spanning Tree (MST) of the following graph. Write down the edges of the MST in the order in which Prim's algorithm adds them to the MST.

1. (A, C), (C, J), (C, D),
(D, B), (B, H), (H, I),
(H, G), (J, E), (E, F)





2. (1) Execute Dijkstra's algorithm on the following graph starting at vertex A. If there are any ties, the vertex with the lower-alphabetic-order letter comes first.

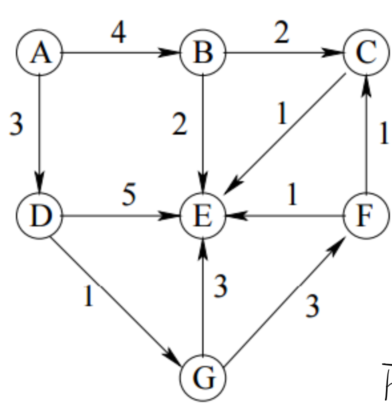
1) List the vertices in the order in which they are deleted from the priority queue and for each the shortest distance from A to the vertex.

2) Draw the shortest paths tree that results.

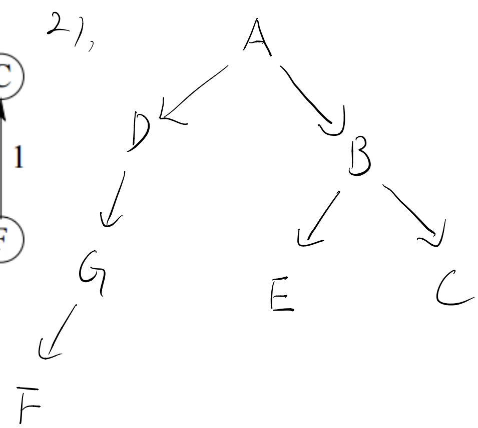
1), A D B G C E F

2, (1)

A	B	C	D	E	F	G
0	∞	∞	∞	∞	∞	∞
0	4	∞	3	∞	∞	∞
0	4	∞	3	8	∞	4
0	4	6	3	6	∞	4
0	4	6	3	6	7	4
0	4	6	3	6	7	4
0	4	6	3	6	7	4
0	4	6	3	6	7	4



2),



(2) Prim and Kruskal Executions

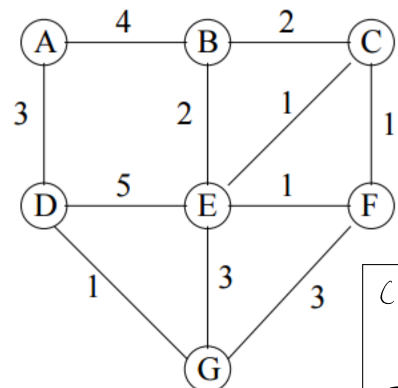
1) Execute Prim's algorithm on the following graph starting at vertex A. If there are any ties, the vertex with the lower letter comes first. List the edges in the order in which they are added to the tree.

2) Execute Kruskal's algorithm on the following graph ~~starting at vertex A~~. Assume that equal weight edges are ordered lexicographically by the labels of their vertices assuming that the lower labeled vertex always comes first when specifying an edge, e.g. (C, E) is before (C, F) which in turn is before (D, G). List the edges in the order in which they are added to the developing forest.

(2), 1),

(A, D), (D, G), (G, E), (E, C), (C, F), (C, B)

A D G E C F B

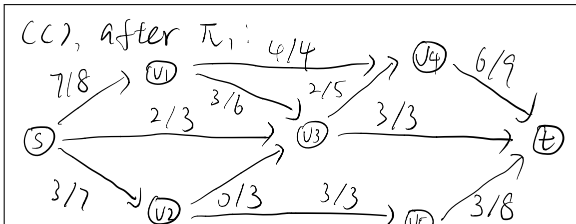


2),

(C, E), (C, F), (D, G), (B, C), (A, D), (E, G)

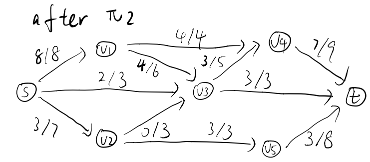
C E F D G B A

(C), after π_1:



3. Consider the flow network N and the flow f shown in the following figure:

after π_2



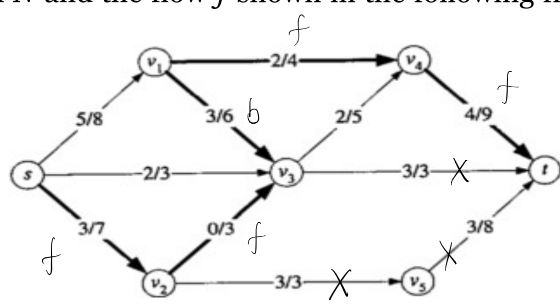
3, (a),

Labeled on figure.

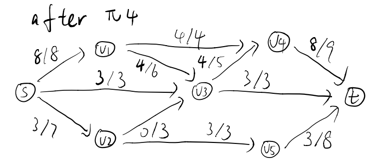
f: forward edges,

b: backward edges.

b要反转一下，整个路径才能通



after π_4



The figure shows an augmenting path π drawn in thick edges.

(b),

不可能的 path 已经用"x"标出，因为这条path，或所有与这条path相邻的path的flow都已经满了。

(a) What are the forward edges of augmenting path π in this figure? What are the backward edges?

(b) How many augmenting paths are there with respect to flow f in this figure? For each such a path, list the sequence of vertices of the path and the residual capacity of the path.

(c) What is the value of a maximum flow in the network N?

π_1: $S \to V_1 \to V_4 \to t$ $\quad \Delta f(\pi_1) = 2$

π_2: $S \to V_1 \to V_3 \to V_4 \to t$ $\quad \Delta f(\pi_2) = 3$

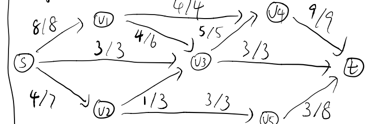
π_3: $S \to V_3 \to V_1 \to V_4 \to t$ $\quad \Delta f(\pi_3) = 1$

π_4: $S \to V_3 \to V_4 \to t$ $\quad \Delta f(\pi_4) = 1$

π_5: $S \to V_2 \to V_3 \to V_1 \to V_4 \to t$ $\quad \Delta f(\pi_5) = 2$

π_6: $S \to V_2 \to V_3 \to V_4 \to t$ $\quad \Delta f(\pi_6) = 3$

after π_6



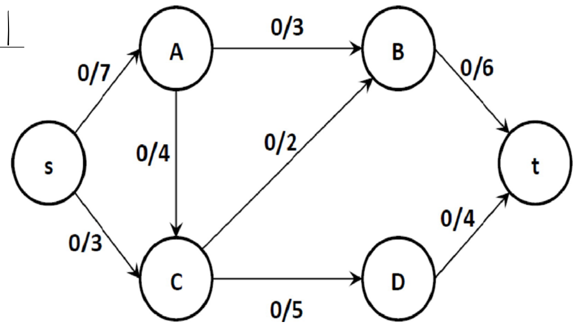
max flow:

$9 + 3 + 3 = 15$

4. Execute the Ford-Fulkerson maximum flow algorithm on the following network:

4,

π	$\Delta f(\pi)$	$\lvert f' \rvert$
$s \to A \to B \to t$	3	3
$s \to A \to C \to B \to t$	2	5
$s \to A \to C \to D \to t$	2	7
$s \to C \to D \to t$	2	9



For each step of the algorithm, write in a table the augmenting path by listing its vertices, its residual capacity and the resulting flow on the network.

5. Consider the following flow network and feasible flow f.

5, (1),

$f = 9+19+9 = 9+14+14 = 37$

(2),

$s \to 3 \to 6 \to 2 \to 5 \to t$

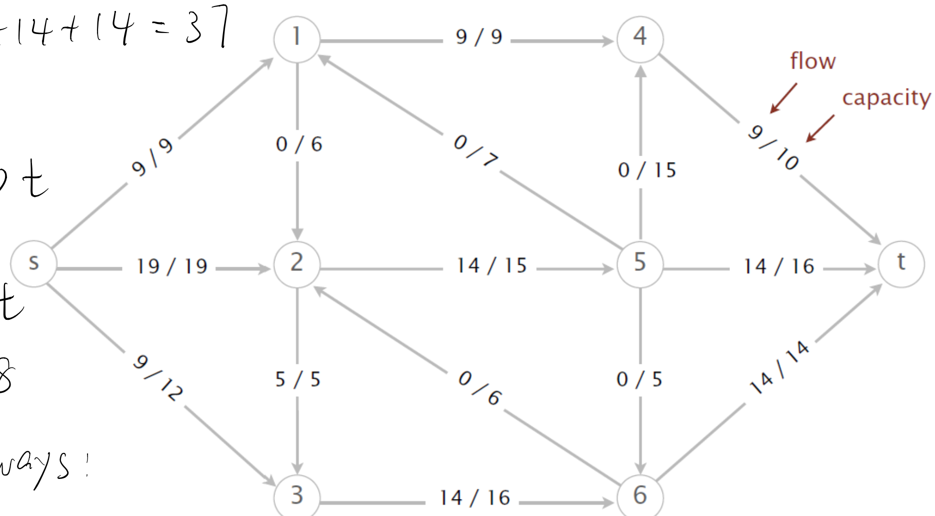
(3), after

$s \to 3 \to 6 \to 2 \to 5 \to t$

max flow is 38

(4), 2 possible ways:

shown as below:



(1) What is the value of the flow f?

(2) Perform one iteration of the Ford-Fulkerson algorithm, starting from the flow f. Give the sequence of vertices on the augmenting path.

(3) What is the value of the maximum flow?

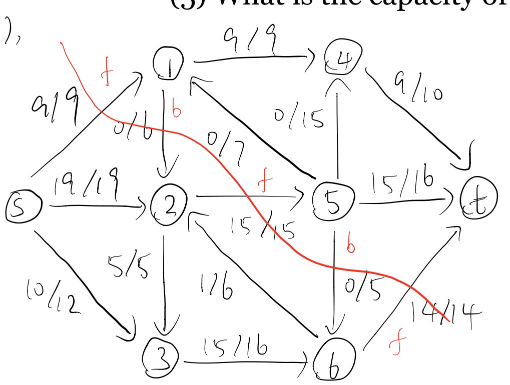
(4) List the vertices on the s side of the minimum cut.

(5) What is the capacity of the minimum cut?

(5), $C(X) = |f|$

$C(X) = 38$

(i),



(ii),