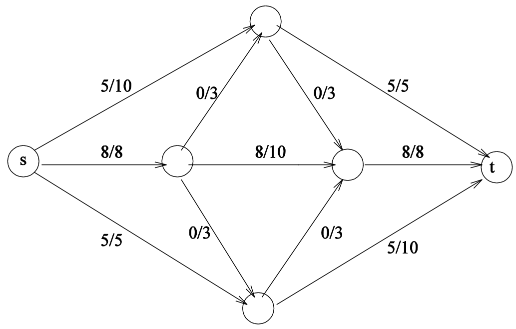
CSC 3190 Introduction to Discrete Mathematics and Algorithms

Assignment 3

Deadline : December 10, 2020 (Thursday) 5:00pm

1 Construct a directed graph G with five vertices and with at most one edge from one vertex to another without self-loop such that the outgoing and incoming degrees of the vertices are (2, 1), (1, 2), (3, 1), (0, 3) and (2, 1) respectively. Show how we can construct G by solving a corresponding network flow problem.

2. The following figure shows a flow network in which an s-t flow has been computed. The label next to each edge represents the flow/capacity on that edge.



(a) What is the value of this flow? Is this a maximum s-t flow? Why?

(b) Find a minimum s-t cut in this flow network, and what is its cut size?

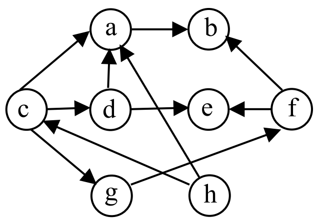
3. Consider the longest increasing subsequence problem:

    Given a sequence S of integers, find the longest increasing subsequence t in S in which the numbers are increasing, i.e., $t[i] < t[i+1]$ for all $i \geq 0$.

(a) Give a pseudo-code to solve the above problem directly with dynamic programming.

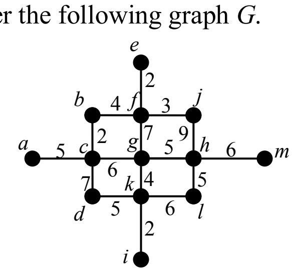
(b) What is the complexity of the algorithm you gave in part (a)?

4. (a) Give a topological sort of the graph below:



(b) Give a pseudo-code to find all the topological sorts of an acyclic graph G.

5. Consider the following graph G.



(a) Construct a minimum spanning tree using the Kruskal's algorithm. Show your steps.

(b) Construct a minimum spanning tree using the Prim's algorithm starting with the node a. Show your steps.

1 Construct a directed graph G with five vertices and with at most one edge from one vertex to another without self-loop such that the outgoing and incoming degrees of the vertices are (2, 1), (1, 2), (3, 1), (0, 3) and (2, 1) respectively. Show how we can construct G by solving a corresponding network flow problem.

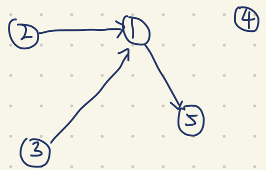
2,1 1,2 3,1

0,3

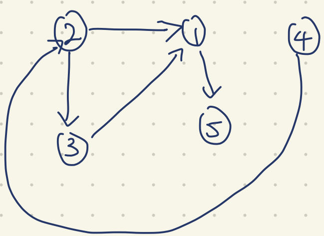
2,1

let 5 vertice be 1, 2, 3, 4, 5

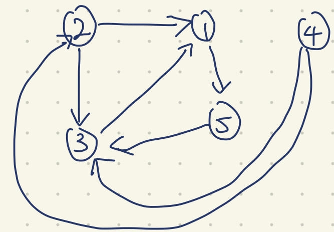
For vertex 1, indegree is 2 & outdegree is 1



For vertex 2, indegree is 1, out degree is 2



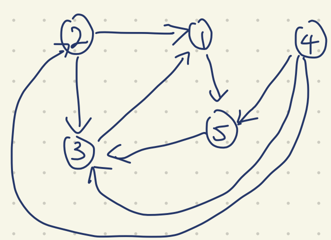
For vertex 3, indegree is 3 out degree is 1



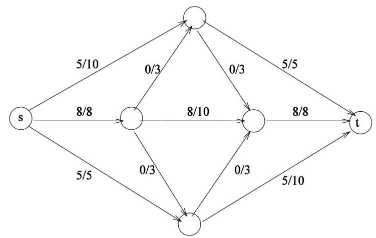
For vertex 4, indreegree is 0 outdregree is 3

& For vertex 5, indegree is 2 outdgee is 1

Final graph:

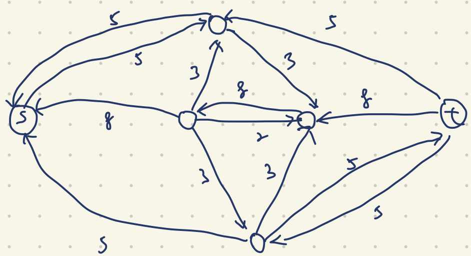


2. The following figure shows a flow network in which an s-t flow has been computed. The label next to each edge represents the flow/capacity on that edge.

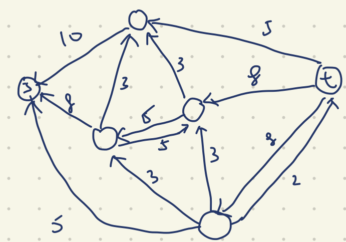


(a) What is the value of this flow? Is this a maximum s-t flow? Why?
(b) Find a minimum s-t cut in this flow network, and what is its cut size?

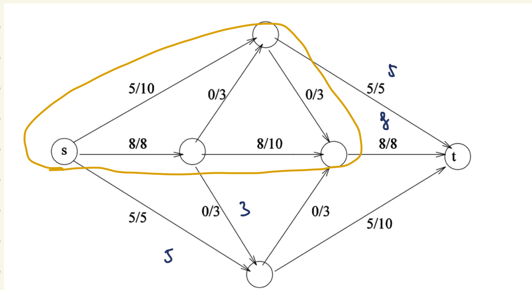
a) The flow is $5+8+5 = 18$



The maximum slow → is 21



b) minimum s-t cut suppose equal to maximum flow (21)



The minimum cut size is $5+8+3+5=21$

3. Consider the longest increasing subsequence problem:

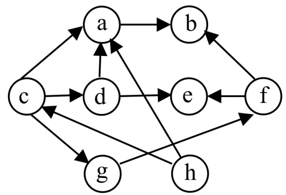
Given a sequence S of integers, find the longest increasing subsequence t in S in which the numbers are increasing, i.e., $t[i] < t[i+1]$ for all $i \geq 0$.

(a) Give a pseudo-code to solve the above problem directly with dynamic programming.
(b) What is the complexity of the algorithm you gave in part (a)?

a)
```
lis (int arr[1.... n])
{ if n == 0 return 0
  m = 1
  for i = 1 to n-1
      if Arr[i] < Arr[n] then
          m = max ( m, 1+ lis [ Arr (a.... i) ]
  return m
}
Main
lis( arr[1...n] )
```

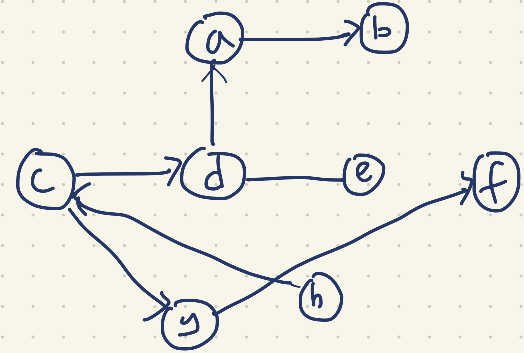
b) $O(n^2)$

4. (a) Give a topological sort of the graph below:



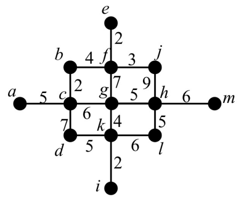
(b) Give a pseudo-code to find all the topological sorts of an acyclic graph *G*.

a)



b)

5. Consider the following graph G.

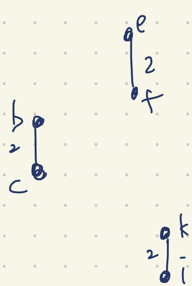


(a) Construct a minimum spanning tree using the Kruskal's algorithm. Show your steps.
(b) Construct a minimum spanning tree using the Prim's algorithm starting with the node a. Show your steps.

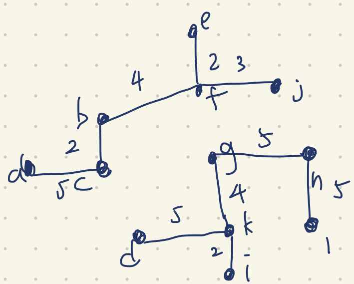
a)

weight		
2	i	k
2	e	f
2	b	c
3	f	j
4	b	f
4	g	k
5	a	c
5	d	k
5	g	h
5	h	l
6	c	g
6	h	m
6	k	l
7	f	g
7	c	d
9	j	h

① pick i,k , e,f , b,c
no cycle form

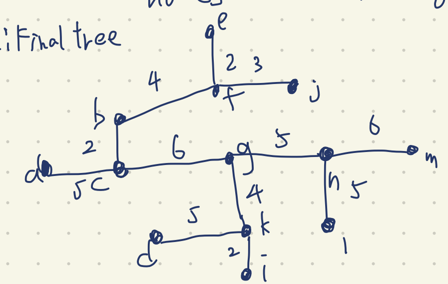


② pick (f j), (b f), (g k), (a, c) , (d,k) , (g, h)
no cycle form



③ pick (c, g) , (h, m)
no cycle form & no of edge = V-1

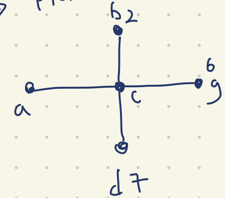
∴ Final tree



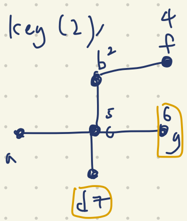
b) Pick vertex a as start point

→ pick vertex with minimum key (5)

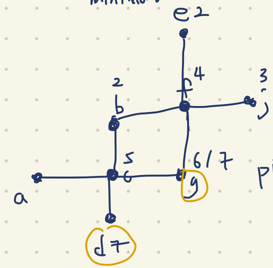
→ pick vertex with minimum key (2),



minikey (4)

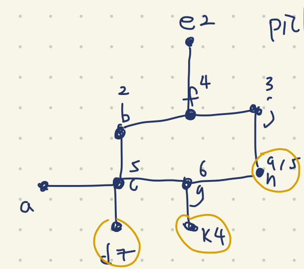


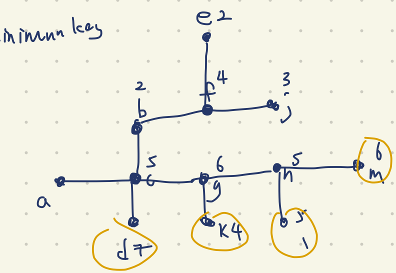
minimum key 2



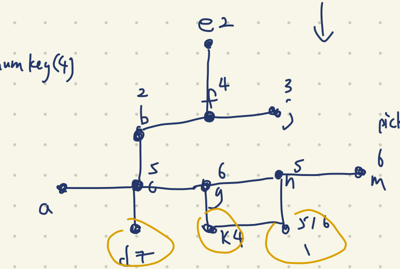
pick 6 (minimum key)

pick 5 (minimum key

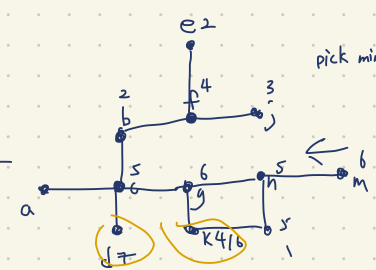




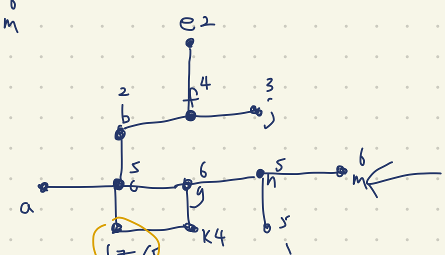
pick 5



pick minmum key(4)



pick minimum key 4



Final graph: