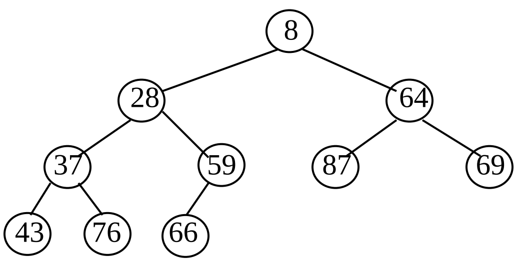
ECS 60 **Preparation for Midterm #2**

Topics: ADTs, heaps, and graphs.

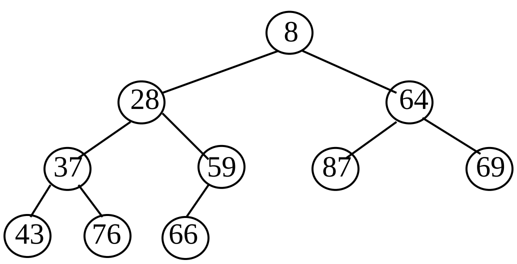
1. (0-25 points) Programs Problem. This question will attempt to determine whether you wrote the code for programming assignment #3 and #4. There is no need for you to see sample questions if you actually participated in writing the code.

2. (30 points) Timetest3 Problem. This question will address what you learned writing your paper for timetest3.cpp. Questions could be of the form where you are given a description of a new File5.dat, and you are asked for the expected result for given ADTs compared to one of the original data files. Questions could also involve determining whether you clearly understand the source(s) of the anomalous behavior of the different ADTs.

3. (15 points) Heap Problem. Heap Problem. There will be one problem dealing insert, deleteMin, or BuildHeap operations with a priority queue. For each of the following binary heaps, show the state of heap after operation specified.



Insert(20)



DeleteMin()

4. (~50 points)Graph Problems There will be two graph problems using one of the following formats.

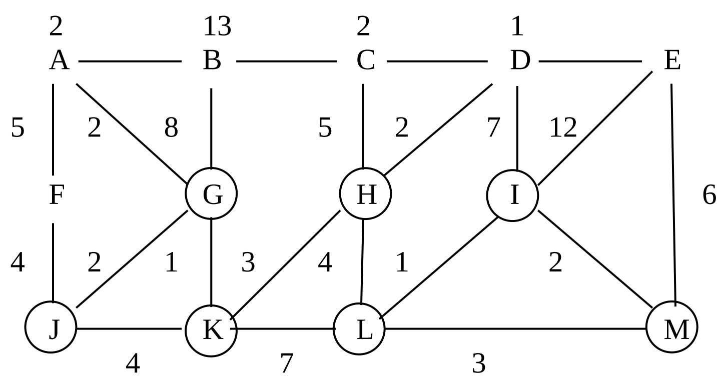
a) Minimum Spanning Tree. Fill in the following tables using Kruskal's algorithm to find the minimum spanning tree of the graph. Use union-by-size, but not path compression. For a union of sets of the same size, the new root should be the old root with the lower array index. Stop the process when you have a minimum spanning tree.

Edge	Weight	Action
(A, B)	1	
(E, H)	2	
(D, G)	3	
(E, F)	4	
(D, F)	5	
(F, H)	6	
(A, C)	7	
(F, G)	8	
(B, C)	9	
(C, F)	10	
(A, H)	11	
(C, H)	12	
(C, G)	13	

Find-Union Array

A	B	C	D	E	F	G	H
0	1	2	3	4	5	6	7

b) Fill in the table using Prim's algorithm to determine the Minimum Spanning Tree from vertex E. The weight of each interior edge is to the left of the edge.



Vertex	known	d_v	p_v
A			
B			
C			
D			
E			
F			
G			
H			
I			
J			
K			
L			
M			

(points) Based on your tables, draw the minimum spanning tree of the graph. (Don't bother with weights.)

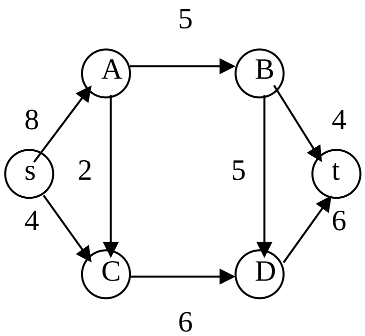
A B C D E

F G H I

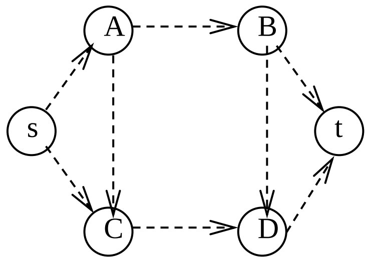
J K L M

c) Network Flow. Choosing the augmenting path that allows the largest increase in flow, find the maximum network flow for the following graph, G. Internal flows are to the left of their respective edges. Your score will be based on your completion of the augmented path line, G_f and G_r , including their initial states and back-flow edges where appropriate.

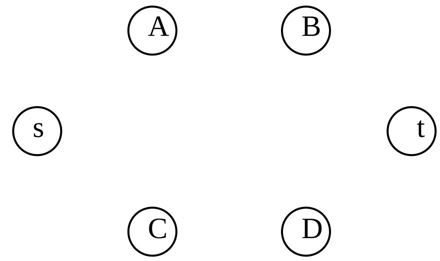
G



G_f

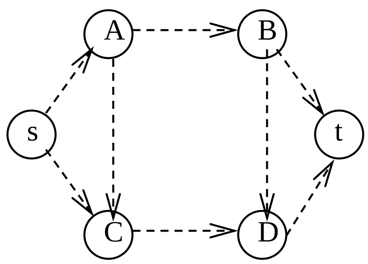


G_r

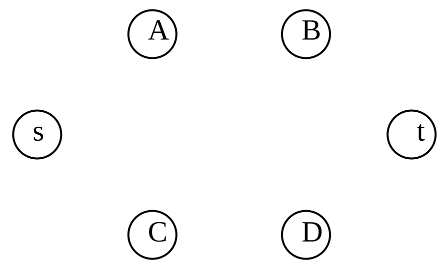


1st augmented path: ________________

G_f

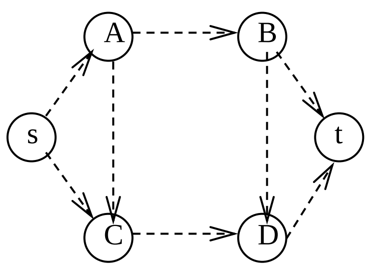


G_r

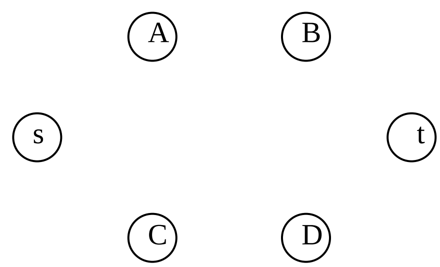


2nd augmented path: ___________________

G_f

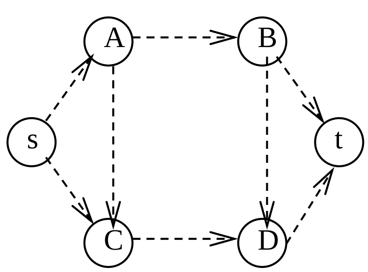


G_r

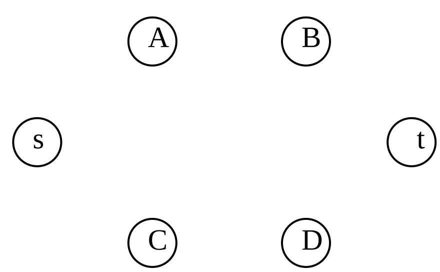


3rd augmented path:___________________

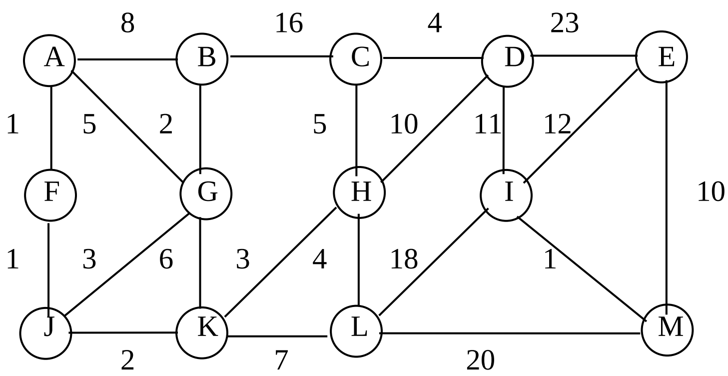
G_f



G_r

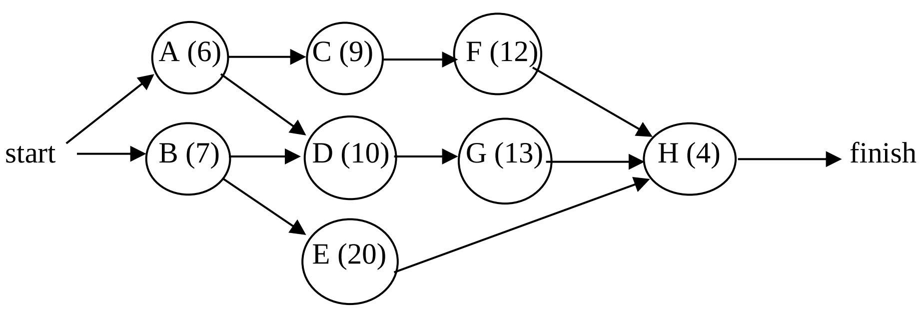


d) Shortest Path. Fill in the table using Dijkstra's algorithm to determine the shortest paths from vertex B The weight of each interior edge is to the left of the edge.



Vertex	known	d_v	p_v
A			
B		**0**	---
C			
D			
E			
F			
G			
H			
I			
J			
K			
L			
M			

e) Event Node Graph. For the following Activity-node graph:

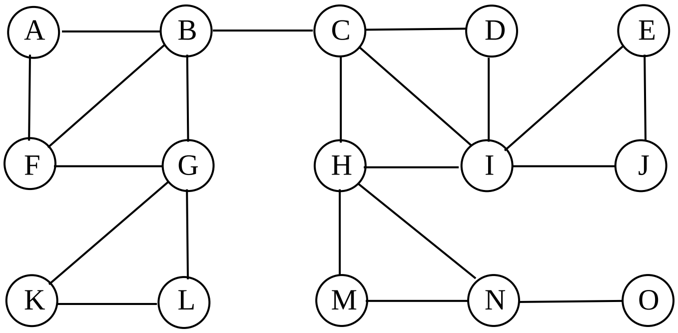


Draw the corresponding Event-node graph using numbered nodes (start numbering at left please).

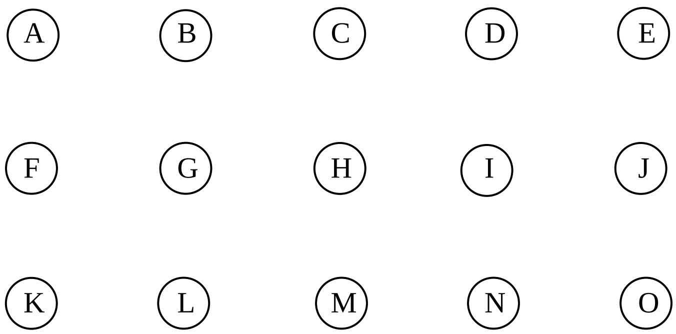
Based on <u>your</u> graph fill in the values for the following table. There may be more columns than needed.

Nodes:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Earliest completion time															
p_v															

f) (35 points) Articulation Points. Using DFS, starting at A, searching in alphabetical order whenever multiple vertices may be processed, give each vertex a number followed by its low. Then list the articulation points



(5 points) Show DFS tree(s) below. You need not show the back edges, but may if you wish.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Vertex number (5 points)															
Low (5 points)															
Articulation Point Y or N (5 points)															

5. ADT and Algorithm Essay (45-50 points) This an ECS 60 problem placed within a familiar context. For the war in Iraq, the Defense Department had to plan how and when to transport troops and materiel to Iraq so that they were all there and operational on March 1st. However, having units arrive earlier than needed would degrade their performance. For each of the 1000 battalions, the planners were given a list of the battalions that it directly required to be operational before that battalion could start setting itself up. It took each battalion 10 days to become operational once ALL of the listed battalions upon which it was dependent were operational.

The planners had 100 ships that took 30 days to travel from the U.S.A. to Kuwait, and 30 days to travel from Kuwait to the U.S.A. Each ship could carry five battalions. How would you approach the planner's problem of scheduling battalion departures? Specifically, describe the ADT(s) and algorithms you would use to determine: 1) The order of departure; and 2) the time of departure of each battalion. Do NOT just say which algorithm(s) you will use. You must explain how you will implement each algorithm within the context of this problem. Here are a some of the questions you need to answer. What are the edges? How are their weights determined? What are the vertices? What determines which vertices are connected to each other? Provide Big-Ohs wherever possible. For the Big-Ohs describe what V, E, N, etc. relate to in this problem. (Note that sorting N items will take NlogN time.)

Algorithm selection : 8 points
Description of graph(s) : 8 points
Graph representation : 2 points
Big–O analysis : 8 points
Explanation of departure ordering process: 12 points
Explanation of departure times process: 12 points