Key

HW5

1. Use DFS to find a topological sort on the following graph. "Mark" each node as Depth First Search would find each node in the following graph. Label each edge as **forward**, back, cross, or tree (f,b,c,t). You may stop early if you discover the graph is not a Dag (explain why). Assume each item is visited in alphabetic order for both restarting DFS and also for visiting edges (start at "a" and visit adjacent nodes that are alphabetically before others as you expand each tree).

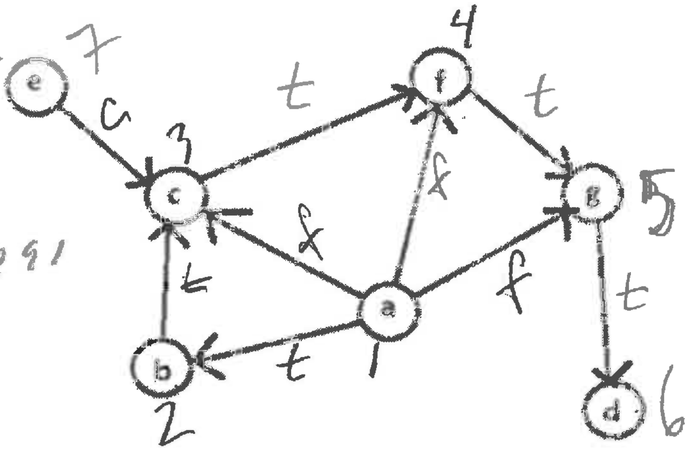
order of dfs return

d, g, f, c, b, a, e

reverse

e, a, b, c, f, g, d

topological sort (not unique)

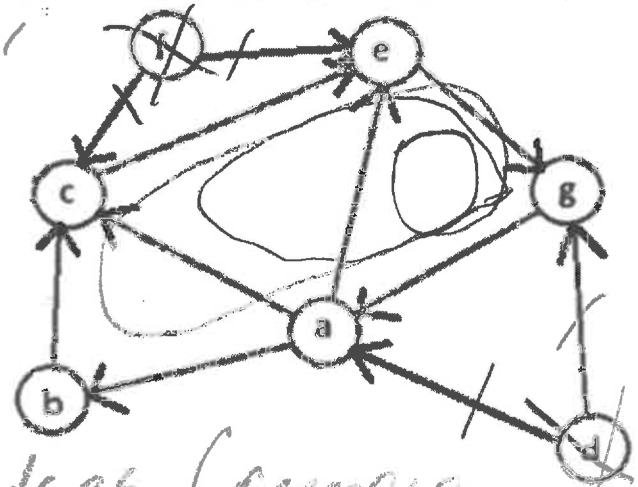


(a,c), (a,f), (a,g) forward b/c mark on 2nd < mark on 1st

(e,c) is cross b/c c is in different tree

2. Use the Source removal to find a topological sort of the following graph. If you cannot do so, then indicate why (start the algorithm and , possibly, stop when you cannot continue or otherwise know the graph is not a Dag). If you cannot give a topological sort, state briefly why.

Remove f & d in either order, stop when can no longer find node w/out any incoming edges (remove outgoing edges of f+d). → If Source removal terminates and there are nodes left : cycle, not a Dag.



3. Perform 5 rounds of Johnson Trotter on the following input. Assume that d>c>b>a, meaning that d will definitely "move" first.

$\overrightarrow{b}\ \overleftarrow{d}\ \overleftarrow{c}\ \overrightarrow{a}$

$\overleftarrow{d}\ \overrightarrow{b}\ \overleftarrow{c}\ \overrightarrow{a}$ (d not mobile, c is)

$\overrightarrow{d}\ \overleftarrow{c}\ \overrightarrow{b}\ \overrightarrow{a}$ — d was mobile max, here

$\overleftarrow{c}\ \overrightarrow{d}\ \overrightarrow{b}\ \overrightarrow{a}$

$\overleftarrow{c}\ \overrightarrow{b}\ \overrightarrow{d}\ \overrightarrow{a}$

$\overleftarrow{c}\ \overrightarrow{b}\ \overrightarrow{a}\ \overrightarrow{d}$

4. Perform Quickselect on the following numbers. You are trying to find the median (middle) value. Use Lomuto Partitioning to partition. Add a piece of paper, use back, or better, use Excel. It is fine to combine the scanning into one line (until one less than pivot is found) and then perform the swap in the next line). Indicate the position of the pivot(p), split(s), and loop counter (i) in your work.

5 2 6 1 7 3 4

ps	i						
5	2	6	1	7	3	4	
p	si						"++s, swap same"
5	2	6	1	7	3	4	
p	s		i				scan 2
5	2	6	1	7	3	4	
p		s	i				"++s, swap"
5	2	1	6	7	3	4	
p		s			i		scan 2
5	2	1	6	7	3	4	
p			s		i		"++s, swap"
5	2	1	3	7	6	4	
p			s			i	scan 1
5	2	1	3	7	6	4	
p				s		i	"++s, swap"
5	2	1	3	4	6	7	
p				s			scan to end (i off end)
5	2	1	3	4	6	7	
p				s			swap p, s
4	2	1	3	5	6	7	
							new round Lomuto, s = 4, need k-1=3
ps	i			x	x	x	x is outside subarray
4	2	1	3	5	6	7	
p	s	i		x	x	x	"++s, swap, ++i"
4	2	1	3	5	6	7	
p		s	i	x	x	x	"++s, swap, ++i"
4	2	1	3	5	6	7	
p			s	x	x	x	"++s, swap, ++i"
4	2	1	3	5	6	7	(i off end)
p			s	x	x	x	swap p, s
3	2	1	4	5	6	7	
s=3, so done							

combining step 1