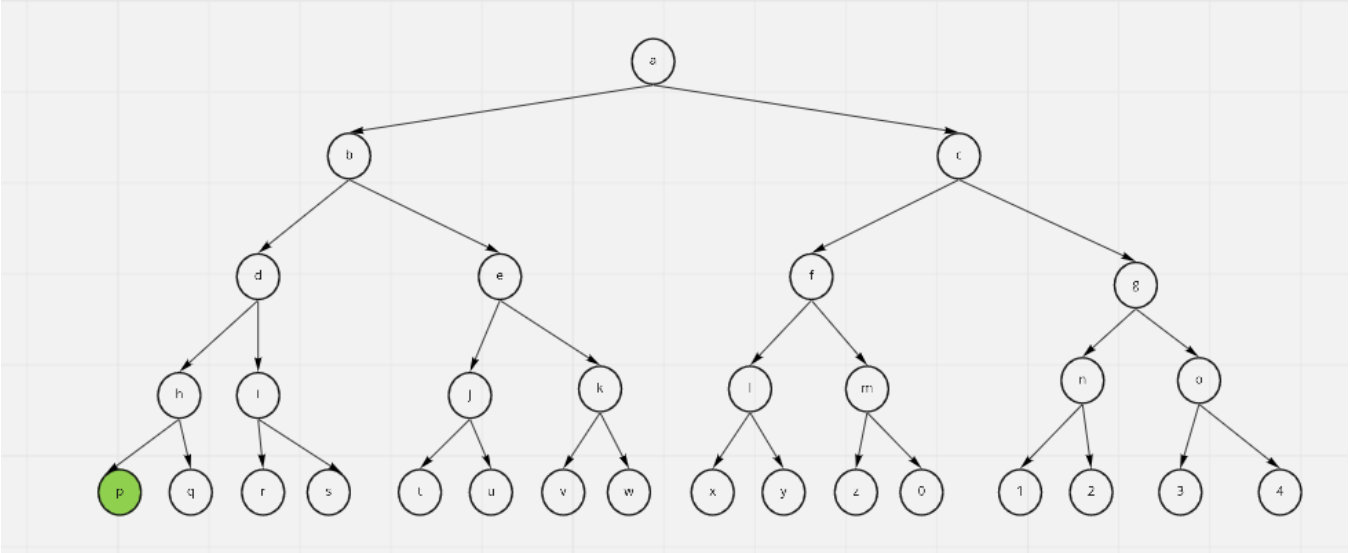
CA318
Labsheet #1

Question 1

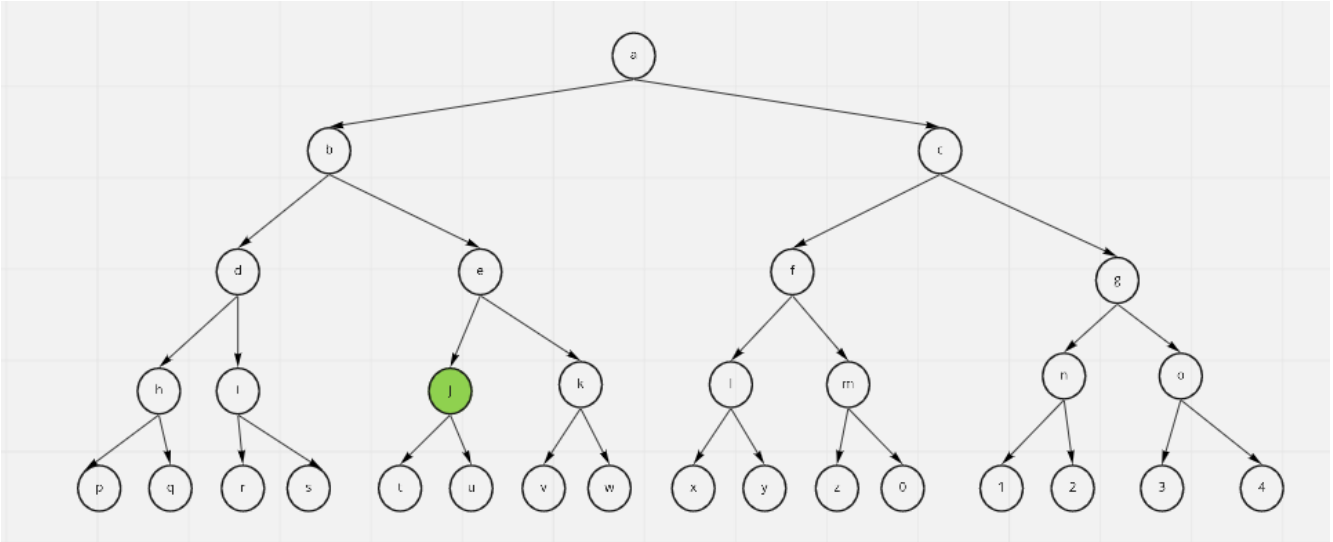
Expand the list of **open** and **closed** nodes in the queue for DFS using the following graphs. A green node shows the goal. Show the workings as the algorithm proceeds down the search tree.

(a)



Closed	Open
a	b,c
a,b	d,e,c
a,b,d	h,i,e,c
a,b,d,h	p,q,i,e,c
a,b,d,h,p	q,i,e,c
STOP	goal p has been reached

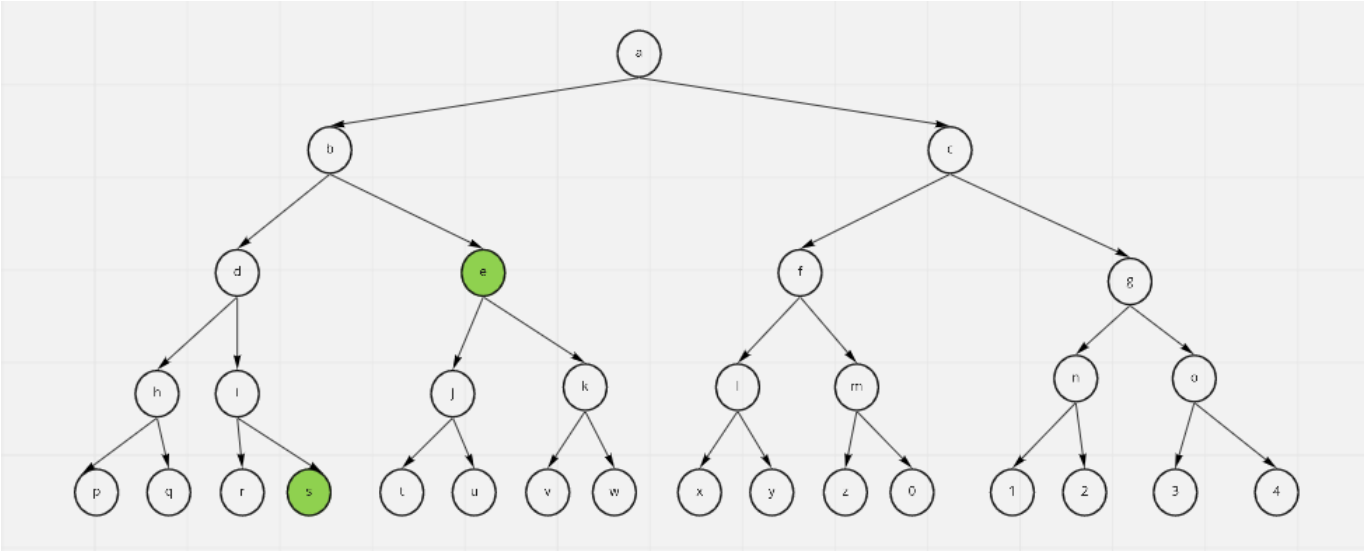
(b)



Closed	Open
a	b,c
a,b	d,e,c
a,b,d	h,i,e,c
a,b,d,h	p,q,i,e,c
a,b,d,h,p	q,i,e,c
a,d,h,p,q	i,e,c [backtrack to i]
a,d,h,p,q,i	r,s,e,c
a,d,h,p,q,i,r	s,e,c
a,d,h,p,q,i,r,s	e,c [backtrack to e]
a,d,h,p,q,i,r,s,e	j,k,c
a,d,h,p,q,i,r,s,e,j	k,c
STOP	Goal j has been reached

Question 2

Consider the following graph, with equal goals at **e** and **s** (in green)



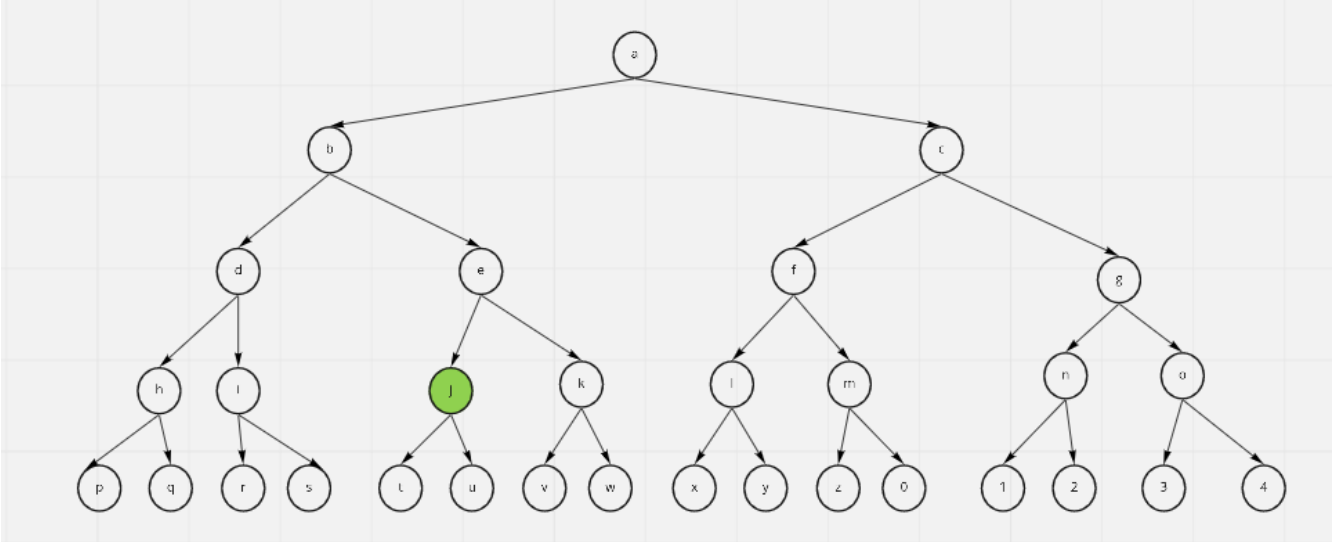
1. Which path to the goal does DFS evaluate first?
    a. [a,b,d,i,s]
2. Describe in your own words how well DFS works in finding the goal (green) in this example
    a. In finds the goal in more steps than BFS would - it misses the shorter path [a,b,e]
3. Can you suggest a faster approach to finding the goal?
    a. BFS would find the shorter path by the evaluating [a,b,c,d,e]

Question 3.

- In order to improve the space complexity of DFS algorithm what kind of changes could we make to the algorithm?
    - We could remove the **closed** list...why do we need it anyway?
- What would be the consequences of the changes you proposed?
    - + It would save space
    - - It would result in expansions of already visited nodes **if we had a graph where a node could be reached by two or more distinct paths**

Question 4.

Looking again at this search tree, what are the **final** open and closed states for BFS at the point where it finds the goal j (green node)?



Closed	Open
a,b,c,d,e,f,g,h,i,j	k,l,m,n,o