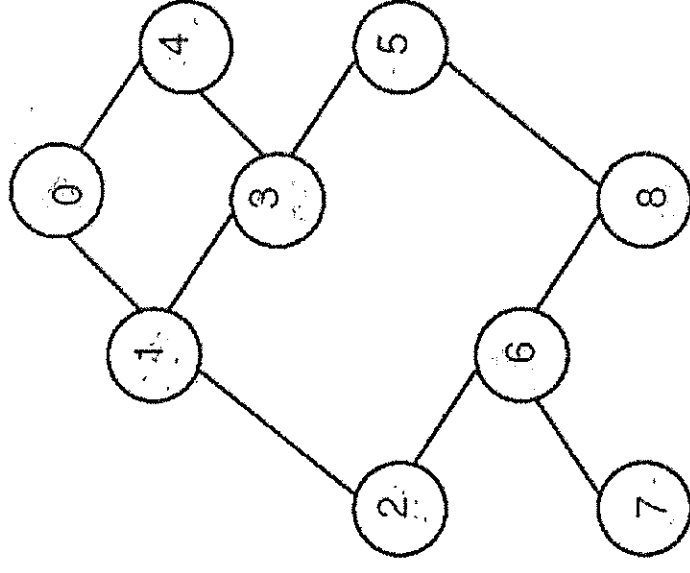


A depth first search traversal solution

```
def dfs(graph, v):  
    visited = []  
    S = [v]  
    while S != []:  
        v = S.pop()  
        if v not in visited:  
            visited.append(v)  
            for w in graph[v]:  
                if w not in visited:  
                    S.append(w)  
    return visited
```



visited	S	v	graph[v]
[]	[]	0	1,4
[0]	[4,4]	0	1,4
[0,1]	[4,2,3]	1	2,3
[0,1,3]	[4,2,4,5]	3	4,5
[0,1,3,5]	[4,2,4,8]	5	8
[0,1,3,5,8]	[4,2,4,6]	8	6
[0,1,3,5,8,6]	[4,2,4,2,7]	6	7,8
[0,1,3,5,8,6,7]	[4,2,4,2]	7	
[0,1,3,5,8,6,7,2]	[4,2,4]	2	8
[0,1,3,5,8,6,7,2,4]	[4,2]	4	8
"	[4]	2	8
"	[]	4	8

0,1,3,5,8,6,7,2,4

0,4,3,1,2,6,8,5,7

0,4,3,5,8,6,7,2,1

