

Practice Problems Set 10

Fall 25

1. Binary Tree Representation (adapted from CLRS Exercise 20.1-2)

A complete binary tree on 7 vertices is given. The vertices are named from **a** to **g** as in a binary heap (i.e., *a* is the root, *b* and *c* are children of *a*, *d* and *e* are children of *b*, etc.). Assume that the edges are undirected.

(a) Give the equivalent adjacency-list and adjacency-matrix representations.

Solution:

Adjacency list (written in Python dictionary/list format):

```
Adj = {  
    'a': ['b', 'c']  
    'b': ['a', 'd', 'e']  
    'c': ['a', 'f', 'g']  
    'd': ['b']  
    'e': ['b']  
    'f': ['c']  
    'g': ['c']  
}
```

Adjacency matrix:

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	0
b	1	0	0	1	1	0	0
c	1	0	0	0	0	1	1
d	0	1	0	0	0	0	0
e	0	1	0	0	0	0	0
f	0	0	1	0	0	0	0
g	0	0	1	0	0	0	0

(b) Give the order of vertex visit (sequence of vertices) starting from the root **a** for both Breadth-First Search (BFS) and Depth-First Search (DFS). While performing each search, visit the outgoing neighbors of a vertex in alphabetical order. For each search, list vertices in the order in which they were first visited.

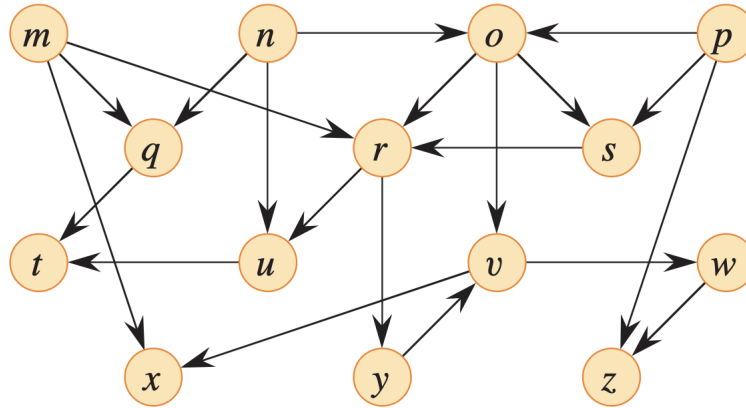
Solution:

Traversal orders:

- BFS: [a, b, c, d, e, f, g]
- DFS: [a, b, d, e, c, f, g]

2. Topological Sort Practice (adapted from CLRS Exercise 20.4-1)

Show the ordering of vertices produced by **TOPOLOGICAL-SORT** when it is run on the DAG below. Assume that the DFS procedure considers the vertices in alphabetical order, and assume that each adjacency list is ordered alphabetically.



Solution:

The depth-first search produces the following *Discovery* and *Finish* times:

	<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>D</i>	1	21	22	27	2	6	23	3	7	10	11	15	9	12
<i>F</i>	20	26	25	28	5	19	24	4	8	17	14	16	18	13

The topologically sorted order is *p, n, o, s, m, r, y, v, x, w, z, u, q, t*.