Jonathan Tso
HW4
CS251

1. a) $E = \{$ (a,b), (a,d), (b,c), (b,f), (c,d), (d,e), (e,f), (f,a), (e,b) $\}$ 9 edges

b)

vertex	indegree	outdegree
a	1	2
b	2	2
c	1	1
d	2	1
e	1	2
f	2	1

c)

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

d)
a → b → d
b → c → f
c → d
d → e
e → b → f
f → a

e) Distinct paths from e to d
1. e, b, c, d
2. e, f, a, d
3. e, f, a, b, c, d
4. e, b, f, a, d

f)
a) 3 length for shortest
b) e, b, c, d
e, f, a, d

g) end/start with d
a) 4
b) d, e, f, a, b, c, d
d, e, b, c, d
d, e, f, a, d
d, e, b, f, a, d

2.

initial graph

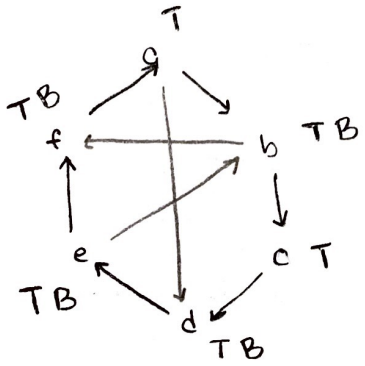
A. Discovered in order:

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

e, b, c, d, f, a

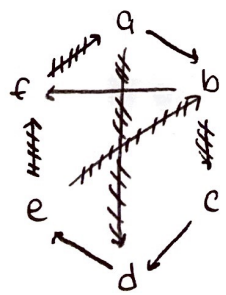
assuming we don't show vertices when we call them back recursively

B.



3. A. Vertex Labels

	shortest-path length	predecessor
a	2	f
b	1	e
c	2	b
d	3	a
e	0	none
f	1	e



5. If a given non-empty directed graph has at least 1 sink, 1 source, it must be a DAG.

↑
FALSE

Consider the above graph, where we have a source a and sink d. We have an intermediate vertex c that cycles back to vertex b. Because of this, it is not acyclic and as a result is not a DAG, thus proving the claim to be false.

4.

A.

a, b, c

a, d, c

B.

Adjacency list 1

a → b → d

b → c

c

d → c

Adjacency list 2

a → d → b

b → c

c

d → c

Source vertex a:
destination: c

<a>
<a, d>
<a, c>
<c>

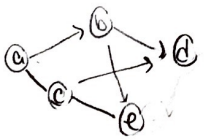
Source vertex a:
destination: c

<a>
<a, b>
<a, c>
<c>

C. $G = (V, E)$ source $s \in V$

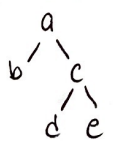
C.1. To show the above statement is false, I will construct a graph $G = (V, E)$ and specifig a source $s \in V$ such that we will have multiple nodes to the destinations d and e and both destinations cannot show their shortest paths for a certain tree, thus no matter the adjacency list, we cannot use BFS to represent it.

C.2.

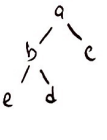


<a>
<a, e>
<a, d, e>
<d, e>

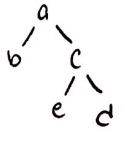
<a>
<a, b>
<a, d, e>
<d, e>



<a>
<a, c>
<e, d>
<e, d>



<a>
<a, b>
<a, e, d>
<e, d>



Consider the possible shortest path tree:

This cannot be represented by a BFS approach, no matter how we set the adjacency list.