

Problem 1. (20 points)

Binary relations may possess some combination of the following properties: Reflexive, Symmetric, Antisymmetric, and Transitive.

For each relation below, list which of the properties each relation possesses.

All relations are defined on the set $S = \{1, 2, 3, 4\}$.

$$a. A = \{(\cancel{1,1}), (1,2), (\cancel{2,2}), (2,3), (3,2), (\cancel{3,3}), (\cancel{4,4})\}$$

$$b. B = \{(\cancel{1,1}), (1,4), (2,3), (3,2), (4,1), (\cancel{4,4})\}$$

$$c. C = \{(2,1), (\cancel{2,2}), (3,2), (\cancel{3,3}), (4,3), (\cancel{4,4})\}$$

$$d. D = \{(\cancel{2,2}), (2,3), (2,4), (3,2), (\cancel{3,3}), (3,4)\}$$

- a)
- a. Reflexive
 - b. Not Symmetric - missing (2,1)
 - c. Not Antisymmetric - has (2,3) and (3,2)
 - d. Not Transitive - missing (1,3)
- b)
- a. Not reflexive - missing (2,2) and (3,3)
 - b. Symmetric
 - c. Not Antisymmetric - has (1,4) and (4,1) & (2,3) and (3,2)
 - d. Not Transitive - missing (2,2) and (3,3)
- c)
- a. Not Reflexive - missing (1,1)
 - b. Not Symmetric - missing (1,2), (2,3), (3,4)
 - c. Antisymmetric
 - d. Not Transitive - missing (4,2)
- d)
- a. Not Reflexive - missing (1,1) and (4,4)
 - b. Not Symmetric - missing (4,2) and (4,3)
 - c. Not Antisymmetric - has (2,3) and (3,2)
 - d. Transitive

Problem 2. (20 points)

For parts a to d below, all relations are defined on the set $S = \{1, 2, 3, 4\}$.

a. List the new pairs that need to be added to relation A to give the **reflexive** closure of A.

$$A = \{(\underline{1,1}), (1,2), (2,1), (\underline{2,2}), (\underline{3,3}), (3,4), (4,3)\}$$

b. List the new pairs that need to be added to relation B to give the **symmetric** closure of B.

$$B = \{(\cancel{1,1}), (\cancel{1,2}), (\cancel{1,3}), (\cancel{2,1}), (\cancel{2,2}), (\cancel{3,1}), (\cancel{3,2})\}$$

c. List the new pairs that need to be added to relation C to give the **transitive** closure of C.

$$C = \{(\cancel{1,1}), (\cancel{2,2}), (2,3), (\cancel{3,3}), (3,4), (\cancel{4,4})\}$$

d. List the new pairs that need to be added to relation D to give the **reflexive transitive** closure of D.

$$D = \{(\cancel{1,1}), (\cancel{1,2}), (\cancel{1,3}), (\cancel{1,4}), (\cancel{2,3}), (\cancel{2,4}), (\cancel{3,3}), (\cancel{3,4}), (\cancel{4,4})\}$$

2,2

- a) (4,4)
- b) (2,3)
- c) (2,4)
- d) (2,2)

Problem 3. (20 points)

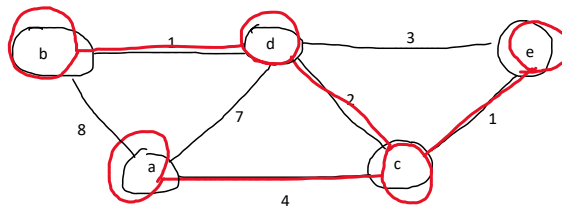
Simulate running Prim's algorithm on the undirected graph represented by the adjacency matrix below.

(Dashes in the matrix represent no edge or no connection.)

(Numbers in the matrix represent edges with weight given by the number.)

Adjacency Matrix

	a	b	c	d	e
a	-	8	4	7	-
b	8	-	-	1	-
c	4	-	-	2	1
d	7	1	2	-	3
e	-	-	1	3	-



Use node 'd' as the initial node in the spanning tree.

Identify the edge added to the spanning tree at each step of the algorithm.

(Remember that an edge is identified by giving a pair of nodes such as {a,b}, rather than by giving the weight on the edge such as 8.)

Step	Nodes	Edges
	(d)	-
a)	b	d,b
b)	c	d,c
c)	e	c,e
d)	a	c,a

Mark	a	b	c	d	e
d	7	<u>1</u>	2	-	3
b	7	<u>1</u>	<u>2</u>	-	3
c	4	<u>1</u>	<u>2</u>	-	<u>1</u>
e	<u>4</u>	<u>1</u>	<u>2</u>	-	<u>1</u>
a	-	-	-	-	-

- Which edge is added on the first step?
- Which edge is added on the second step?
- Which edge is added on the third step?
- Which edge is added on the fourth step?

- d,b
- d,c
- c,e
- c,a

Problem 4. (20 points)

Simulate running Dijkstra's algorithm on the undirected graph represented by the adjacency matrix below.

(Dashes in the matrix represent no edge or no connection.)

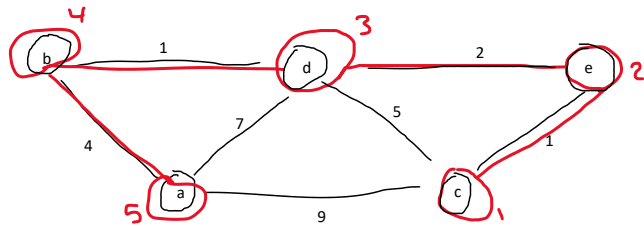
(Numbers in the matrix represent edges with weight given by the number.)

Use node 'c' as the source node.

Follow parts a through d below to give the distance values assigned to each node on each step of the algorithm.

Adjacency Matrix

	a	b	c	d	e
a	-	4	9	7	-
b	4	-	-	1	-
c	9	-	-	5	1
d	7	1	5	-	2
e	-	-	1	2	-



The initial distance values assigned to the nodes are:

(Dashes represent infinite distance.)

Initial Distance Values

a	b	c	d	e
-	-	0	-	-

part	mark	a	b	c	d	e
		-	-	0	-	-
a)	c	9	-	0	5	1
b)	e	9	-	0	3	1
c)	d	9	4	0	3	1
d)	b	8	4	0	3	1
	a					

a. Give the distance values assigned to each node after the first node is marked and the distances are updated.

b. Give the distance values assigned to each node after the second node is marked and the distances are updated.

c. Give the distance values assigned to each node after the third node is marked and the distances are updated.

d. Give the distance values assigned to each node after the fourth node is marked and the distances are updated.

part	mark	a	b	c	d	e
a)	c	9	-	0	5	1
b)	e	9	-	0	3	1
c)	d	9	4	0	3	1
d)	b	8	4	0	3	1

Problem 5. (20 points)

Simulate running Floyd's algorithm on the undirected graph represented by the following distance matrix. (Dashes in the matrix represent infinite distance.)

Distance Matrix

	a	b	c	d	e
a	0	4	9	7	-
b	4	0	-	1	-
c	9	-	0	5	1
d	7	1	5	0	2
e	-	-	1	2	0

a. Give the value stored in the matrix at row 'b' and column 'c' after pivoting with node 'a'.

a) 13

b. Give the value stored in the matrix at row 'd' and column 'a' after pivoting with nodes 'a' and 'b'.

b) 5

c. Give the value stored in the matrix at row 'a' and column 'e' after pivoting with nodes 'a', 'b', and 'c'.

c) 10

d. Give the value stored in the matrix at row 'c' and column 'b' after pivoting with nodes 'a', 'b', 'c', and 'd'.

d) 6

Original

	a	b	c	d	e
a	0	4	9	7	-
b	4	0	-	1	-
c	9	-	0	5	1
d	7	1	5	0	2
e	-	-	1	2	0

Pivot (a)

	a	b	c	d	e
a	0	4	9	7	-
b	4	0	13(a)	1	-
c	9	13(a)	0	5	1
d	7	1	5	0	2
e	-	-	1	2	0

Pivot (a, b)

	a	b	c	d	e
a	0	4	9	5(b)	-
b	4	0	13(a)	1	-
c	9	13(a)	0	5	1
d	5(b)	1	5	0	2
e	-	-	1	2	0

Pivot (a, b, c)

	a	b	c	d	e
a	0	4	9	5(b)	10(c)
b	4	0	13(a)	1	14(c)
c	9	13(a)	0	5	1
d	5(b)	1	5	0	2
e	10(c)	14(c)	1	2	0

Pivot (a, b, c, d)

	a	b	c	d	e
a	0	4	9	5(b)	7(d)
b	4	0	6(d)	1	3(d)
c	9	6(d)	0	5	1
d	5(b)	1	5	0	2
e	7(d)	3(d)	1	2	0

Problem 6. (20 points)

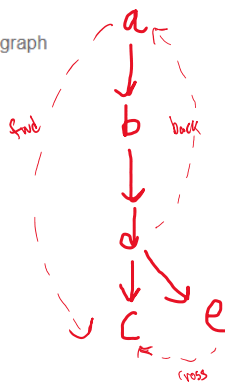
Simulate running Depth-First-Search-Forest on the directed graph represented by the adjacency lists below.

Start the search at node 'a'.

Whenever there is a choice of which node to visit next, choose the nodes in alphabetical order.

Adjacency Lists

a:	b, c
b:	d
c:	
d:	a, c, e
e:	c



a. List the edges of the graph that are 'tree' edges in the resulting DFS-Forest. (Please identify each edge with a pair of letters such as a,b)

a) a,b b,d d,c d,e

b. List the edges of the graph that are 'forward' edges in the resulting DFS-Forest.

b) a,c

c. List the edges of the graph that are 'backward' edges in the resulting DFS-Forest.

c) d,a

d. List the edges of the graph that are 'cross' edges in the resulting DFS-Forest.

d) e,c

Problem 7. (20 points)

Using the following relations:

Relation s

A	B	C
1	2	1
2	1	2
3	3	2

Relation t

B	C	D
1	1	2
1	2	2

a)

B	C
1	1
1	2

b)

A	B	C
2	1	1
2	1	2

project[AB](s)

A	B
1	2
2	1
3	3

project[BC](t)

B	C
1	1
1	2

c)

A	B	C	D
2	1	2	2

s

A	B	C
1	2	1
2	1	2
3	3	2

t

B	C	D
1	1	2
1	2	2

Give the relation that results from each relational algebra expression below. Please label each column in the result with the appropriate attribute name.

a. $\pi_{BC} t$

b. $(\pi_{AB} s) \bowtie (\pi_{BC} t)$

c. $s \bowtie t$

d. $t - (\rho_{A \leftarrow D} s)$

d)

B	C	D
1	1	2

t

B	C	D
1	1	2
1	2	2

rename[A←D](s)

B	C	D
2	1	1
1	2	2
3	2	3

Problem 8. (20 points)

Follow parts a through d below to evaluate the following Datalog Rules using relational algebra operations and the fixed-point algorithm.

Rules:

$a(Y,X) :- a(X,Z), b(Z,Y)$
$b(X,Y) :- b(Z,X), a(Y,Z)$

a

X	Z
1	2
2	3
2	1

b

Z	Y
2	2

X	Z	Y
1	2	2

project

Y	X
2	1

b

Z	X
2	2
2	1

a

Y	Z
1	2
2	3

Y	Z	X
1	2	2
1	2	1

project

X	Y
2	1
1	1

The initial contents of the relations as defined by the Schemes and Facts are as follows:

Relation a

C	D
1	2
2	3

Relation b

A	B
2	2

a. List the tuples added to 'Relation a' by the **first** iteration over the rules. (Use only tuples from previous iterations when using rules to create new tuples.)

a) 2,1

b. List the tuples added to 'Relation b' by the **first** iteration over the rules. (Use only tuples from previous iterations when using rules to create new tuples.)

b) 2,1

c. List the tuples added to 'Relation a' by the **second** iteration over the rules. (Use only tuples from previous iterations when using rules to create new tuples.)

c) none

d. List the tuples added to 'Relation b' by the **second** iteration over the rules. (Use only tuples from previous iterations when using rules to create new tuples.)

d) 1,1