Evan Wilcox
CS1200 Fall 2018
Homework 6
Due: Monday 11/26/18

1. Let R be the relation on A and B defined by aRb iff a divides b.

   (a) List the pairs in R.

   $$R = \{(2, 10), (3, 3), (3, 15), (3, 33), (5, -5), (5, 10), (5, 15)\}$$

   (b) Represent the relation R as a 0,1-matrix.

   $$\begin{array}{cc} & B \\ & \begin{array}{ccccccc} \text{-5} & 3 & 7 & 10 & 15 & 33 & 37 \end{array} \\ A \begin{array}{c} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

2. R is symmetric but is not reflexive, antisymmetric or transitive.

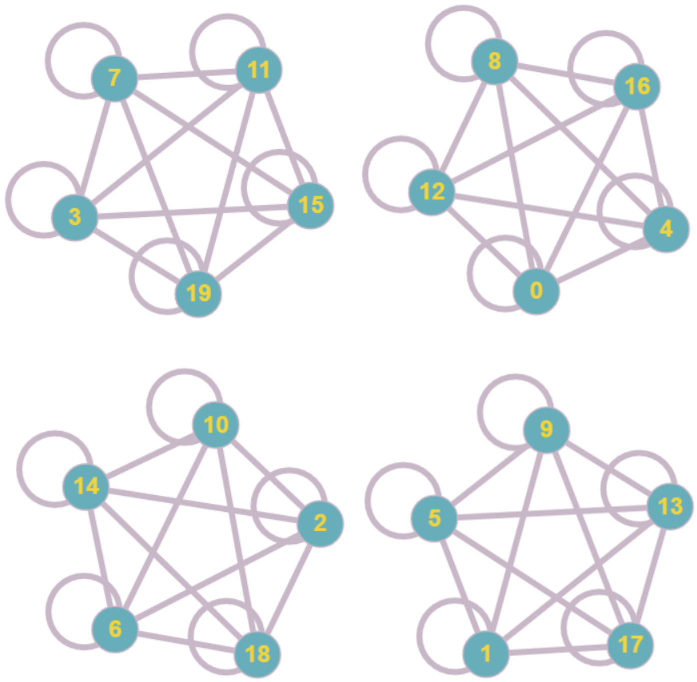
3. (a) R is reflexive, symmetric and transitive but not antisymmetric. R is an equivalence relation but not a well order, partial order or total order.

   (b) Use python functions to verify your answers.

```
PS C:\Users\Evan\Documents\MST\CS1200\HW06> python .\hw06.py
R = [(0, 0), (0, 4), (1, 1), (1, 3), (2, 2), (3, 1), (3, 3), (4, 0), (4, 4)]
Reflexive = True
Symmetric = True
AntiSymmetric = False
Transitive = True
Partial Order = False
Total Order = False
Well Order = False
Equivalence Relation = True
```

```python
#3b
A = [0, 1, 2, 3, 4]
R = [(0, 0),(0, 4),(1, 1),(1, 3),(2, 2),(3, 1),(3, 3),(4, 0),(4, 4)]

def Reflexive(D,R):
    for x in D:
        if (x,x) not in R:
            return False
    return True

def Symmetric(R):
    for (a,b) in R:
        if (b,a) not in R:
            return False
    return True

def AntiSymmetric(R):
    for (a,b) in R:
        if (a !=b) and ((b,a) in R):
            return False
    return True

def Transitive(R):
    for (a,b) in R:
        for (c,d) in R:
            if (b == c) and ((a,d) not in R):
                return False
    return True

def PartialOrder(D, R):
    return Reflexive(D,R) and AntiSymmetric(R) and Transitive(R)

def TotalOrder(D, R):
    for a in D:
        for b in D:
            if (a, b) not in R and (b, a) not in R:
                return False
    return True

def WellOrder(R):
    for (a, b) in R:
        if not (a <= b):
            return False
    return True

def EquivalenceRel(D, R):
    return Reflexive(D,R) and Symmetric(R) and Transitive(R)

print "R =", R
print "Reflexive =", Reflexive(A,R)
print "Symmetric =", Symmetric(R)
print "AntiSymmetric =", AntiSymmetric(R)
print "Transitive =", Transitive(R)
print "Partial Order =", PartialOrder(A, R)
print "Total Order =", TotalOrder(A, R)
print "Well Order =", WellOrder(R)
print "Equivalence Relation =", EquivalenceRel(A, R)
```

4. (a) Draw the directed graph representing R. If R can be represented by an undirected graph, draw it.

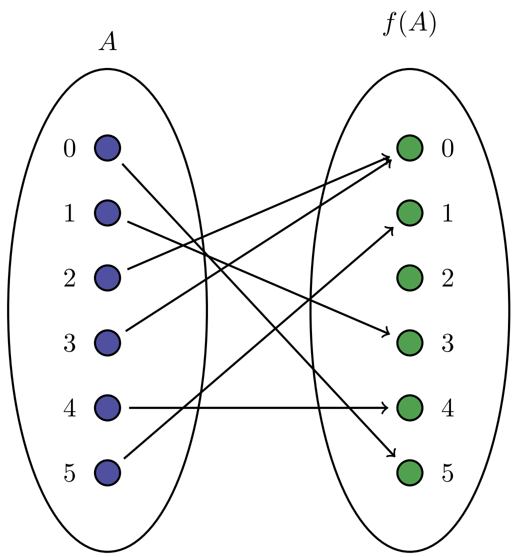


(b) Draw a 0,1-matrix to represent R with the rows and columns labeled.

$$
\begin{array}{cc}
 & B \\
 & \begin{array}{cccccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \end{array} \\
A \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \\ 19 \\ 20 \end{array} &
\begin{bmatrix}
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{bmatrix}
\end{array}
$$

(c) R is an equivalence relation but not a partial order, total order or well order.

5. (a) Draw a bipartite graph representation of f.



(b) f is not an injection, surjection, or bijection.

(c) Write a Python program to test whether the functions are injections, surjections, or bijections.

```python
#5
A = [0, 1, 2, 3, 4, 5]
R = [(0, 5), (1, 3), (2, 0), (3, 0), (4, 4), (5, 1)]

def Injection(R):
    Bs = []
    for (a, b) in R:
        Bs.append(b)

    for x in Bs:
        if Bs.count(x) > 1:
            return False

    return True

def Surjection(A, R):
    Bs = []
    for (a, b) in R:
        Bs.append(b)

    for x in A:
        if x not in Bs:
            return False

    return True

def Bijection(A, R):
    return (Injection(R) and Surjection(A, R))

print "Injection =", Injection(R)
print "Surjection =", Surjection(A, R)
print "Bijection =", Bijection(A, R)
```

```
PS C:\Users\Evan\Documents\MST\CS1200\HW06> python .\hw06.py
Injection = False
Surjection = False
Bijection = False
```

6. (a) Write the formulas for $g \circ f$ and $f \circ g$.

$$g \circ f = 12n + 9$$

$$f \circ g = 12n + 5$$

(b) Determine whether each of $f, g, g \circ f, f \circ g$ are injections, surjections or bijections.

Function	injection?	surjection?	bijection?
f	Yes	Yes	Yes
g	Yes	Yes	Yes
$g \circ f$	Yes	Yes	Yes
$f \circ g$	Yes	Yes	Yes

(c) Compute $(f \circ g)^{-1}(\{-5, -3, 0, 7, 9, 21, 22, 23, 45\})$ and $(g \circ f)^{-1}(\{-7, 0, 5, 7, 9, 17, 22, 41\})$.

n	-5	-3	0	7	9	21	22	23	45
$(f \circ g)^{-1}(n)$	$-\frac{5}{6}$	$-\frac{2}{3}$	$-\frac{5}{12}$	$\frac{1}{6}$	$\frac{1}{3}$	$1\frac{1}{3}$	$1\frac{5}{12}$	$1\frac{1}{2}$	$3\frac{1}{3}$

n	-7	0	5	7	9	17	22	41
$(g \circ f)^{-1}(n)$	$-1\frac{1}{3}$	$-\frac{3}{4}$	$-\frac{1}{3}$	$-\frac{1}{6}$	0	$\frac{2}{3}$	$1\frac{1}{12}$	$2\frac{2}{3}$

7. Let N be the set of natural numbers and $f : N \to N$ be the function $f(n) = 5n + 4$.

n	4	5	8	9	10	11	13	14	15	24	30
$f^{-1}(n)$	0	$\frac{1}{5}$	$\frac{4}{5}$	1	$1\frac{1}{5}$	$1\frac{2}{5}$	$1\frac{4}{5}$	2	$2\frac{1}{5}$	4	$5\frac{1}{5}$