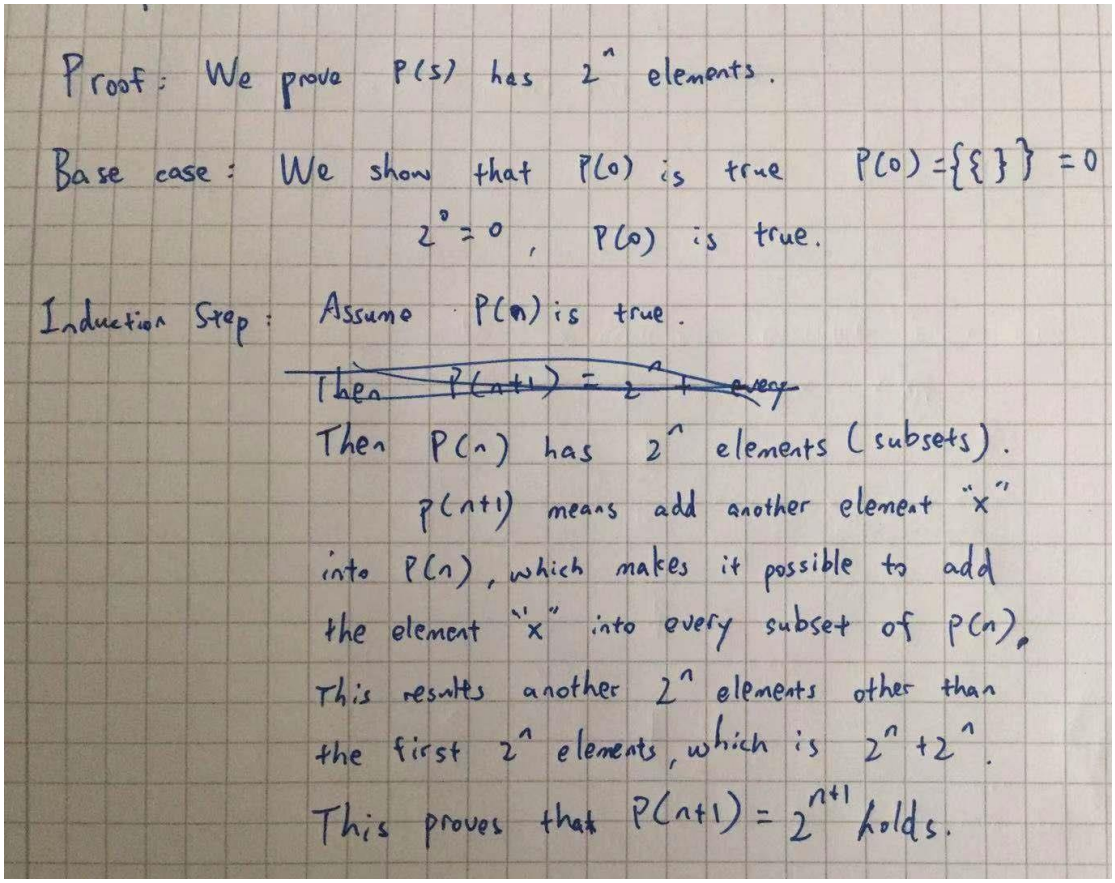
Problem 3.1



Problem 3.2:

a)

Irreflexive. a!=a is false.

Symmetric. a!=b and b!=a both are true.

Not transitive. a!=b and b!=c, a=c can still be a case.

b)

Reflexive. a-a is always 0, which is <=3.

Symmetric. |a-b|<=3 means the distance between a and b is less than or equal to 3. Therefore, |b-1|<=3 does not contradict because for this the distance is also less than or equal to 3.

Not transitive. While the distance between a and b is 3, the distance between b and c is 3, which means the distance betwene a and c can be at most 6.

c)

Reflexive. (a mod 10)=(a mod 10)

Symmetric. (a mod 10)=(b mod 10) can be also written as (b mod 10)=(a mod 10)

Transitive. (a mod 10)=(b mod 10) and (b mod 10)=(c mod 10) can be written as (a mod 10)=(b mod 10)=(c mod 10)

Problem 3.3:

a)

```
isPrime :: Int -> Bool
isPrime x = one x 2
```

```
one :: Int -> Int -> Bool
one x y
  | x == 1 = False
  | x == 2 = True
  | y == x `div`2 = True
  | x `mod` y /= 0 = one x (y+1)
  | x `mod` y == 0 = False

b)
isPrime :: Int -> Bool
isPrime x = one x 2

one :: Int -> Int -> Bool
one x y
  | x == 1 = False
  | x == 2 = True
  | y == x `div`2 = True
  | x `mod` y /= 0 = one x (y+1)
  | x `mod` y == 0 = False

rotate :: Int -> [a] -> [a]
rotate x [] = []
rotate 0 list = list
rotate 1 list = tail list ++ [head list]
rotate x list = rotate (x-1) (tail list ++ [head list])

show :: Show a => a -> String
read :: Read a => String -> a
isCircPrime :: Integer -> Bool
isCircPrime x
  | a == length (show x) = True
  | isPrime x == True = isCircPrime (read (rotate (a+1) (show x) ) :: Integer)
  | isPrime x == False = False
  where a = 0
```