Theory Of Computation Report

Rakshit

2017CSB1101

Question 1: Find Remainder when x is divided by y

Explanation:
(Long Division method is used)
The computation of the remainder is performed in the following manner.

1. Initially, symbols '$' and '^' are appended at the end of the input and at a distance of 'n' bits from $ respectively. The 'n' is the bit length of y.
2. Now the algorithm would enter an infinite loop that will be terminated at particular points.
3. The first 1 present in the x will be searched using find1() subroutine.
4. If it is not found then the algorithm will EXIT out of the loop and would print the remainder.
5. If present then it will search if 'n' bits are present in the x bits from the first 1 using checkbits() subroutine with an input of n.
6. If they are not present then again the algorithm will EXIT out of the loop and would print the remainder.
7. If present then we will compare the bits of x with that of y. For this compareval() subroutine is used which will return the output being 0, 1, 2. Each returned value signifies
   0 : x[first1..n bits] = y
   1: x[first1..n bits] > y
   2: x[first1..n bits] < y
8. For case 0: All the respective bits are turned to 0 and the loop is started again with step 3.
9. For case 1: A copy of y is computed and stored in the area between '$.....^'.
   Now the area of the x will be highlighted using highlightarea() subroutine which takes the input of first1 and length of bits to be highlighted.
   The subtraction is then performed with subtract() subroutine which will take input of first 1, n, and 0.
10. For case 2: Since given x is less than y hence further subtraction can only be performed if we use n+1 bits in x instead of n bits.
    So checkbits() subroutine is again used with values of first 1 and n+1.

If the bits are not present then the algorithm will EXIT out of the loop and would print the remainder.
If they are present then again A copy of y is computed and stored in the area between '$.....^'.
Now the area of the x will be highlighted using highlightarea() subroutine which takes the input of first1 and length of bits to be highlighted i.e n+1.
The subtraction is then performed with subtract() subroutine which will take input of first 1, n, and 1.

11. Go back to step 3.

Highlight() will work as substituting 1 with a and 0 with b.

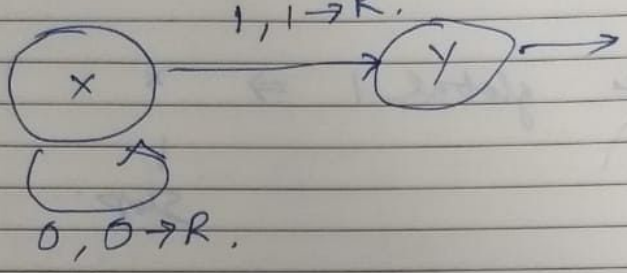
Now the subtract() subroutine will first find the last bit in y' (Copy of y) which is left of end or 'z'.
Based on the bit value it will take route 1 or route 0 while changing that bit to 'z' and going to the last highlighted bit of x.
If the highlighted bit is b and it is on route 1 then it will change that bit to 1 and borrow from the left bits. To borrow another subroutine will be called which is until1() which will change all the left 0 bits to 1 until a left 1 is found which will be converted to 0.
If the highlighted bit is a and it is on route 1 then that bit is changed to 0.
If the highlighted bit is a/b and it is on route 0 then the substitution will be reverted.
Go back to the start of the subroutine. If no such bits are found then the subroutine will exit.

JANUARY

28 Wk 04 • 28th Day
SATURDAY

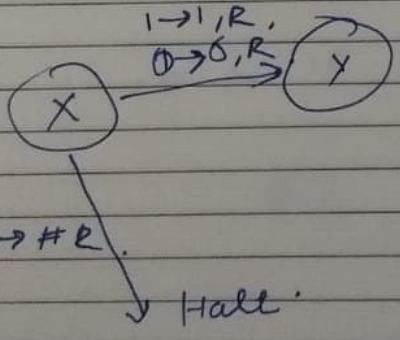
~~First~~

Find 1



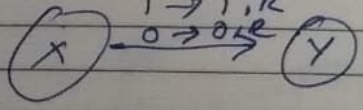
Check bits

n states
for state 1 ... n-1



29 Sunday

for state n th

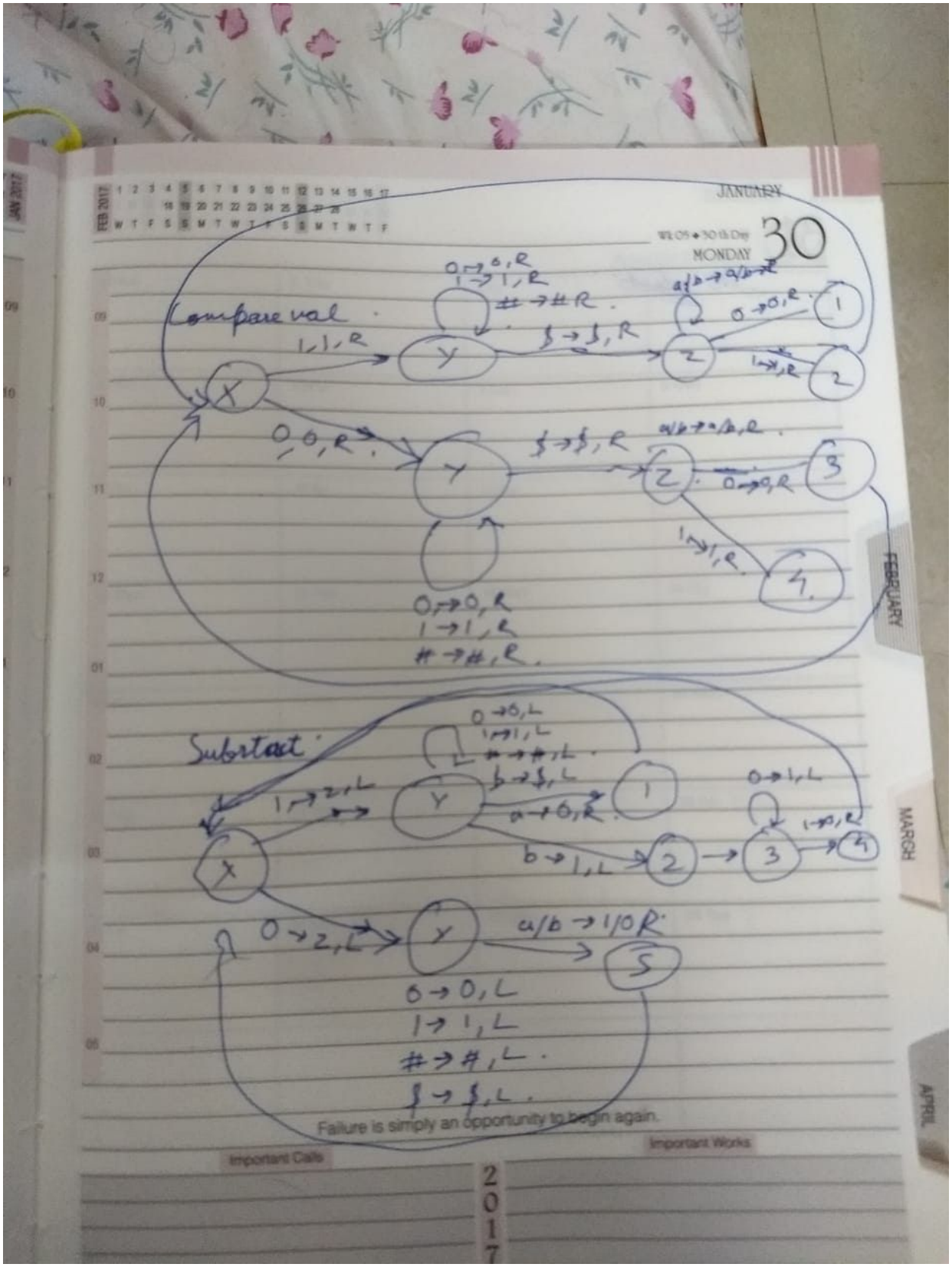


Happiness is not the absence of problems, it's ability to deal with them.

Important Works

Important Calls

2017

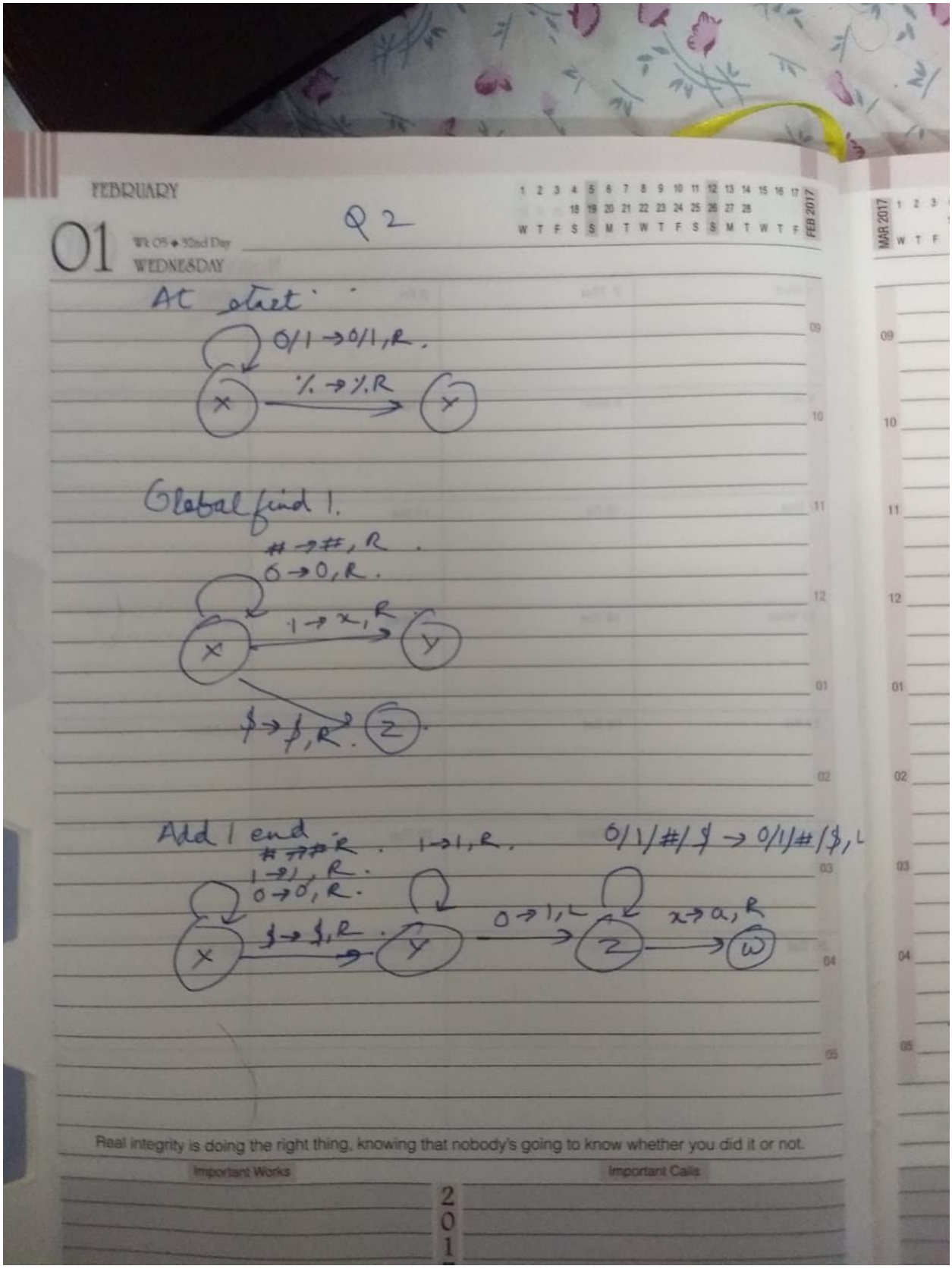

JANUARY
MONDAY 30
Compare val.
Subtract.
Failure is simply an opportunity to begin again.
Important Calls
Important Works
2017

Question 2: Find Max degree

Explanation:
The computation is performed as follows

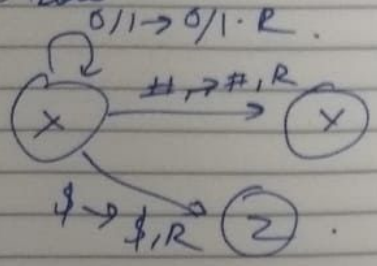
1. The first ‘#’ is substituted by ‘%’ and ‘$’ is appended at the end of the matrix.
2. Now the algorithm would enter an infinite loop that will be terminated at particular points.
3. The pointer will be put to the start of the matrix using the atstart() subroutine.
4. The first 1 present in the matrix will be searched using globalfind1().
5. If it is not found then the algorithm will EXIT out of the loop and the tape will be reset for reverting substitutions and then converting the unary at the end to binary using unarytobinary() subroutine. Then the whole tape will be printed.
6. If the first 1 is found then it is converted to ‘x’. Then add1end() subroutine is called which will add 1 after $ at an empty place and would go back to revert that ‘x’ into ‘a’.
7. Now the nextrow() subroutine is called which will fetch the first position of the next row.
8. If nextrow() is after ‘$’ then we will exit from the loop else we will go into the loop and find the first 1 in the row using subroutine find1() and substituting with ‘a’.
9. Go back to step 7.

10. Go back to step 3.

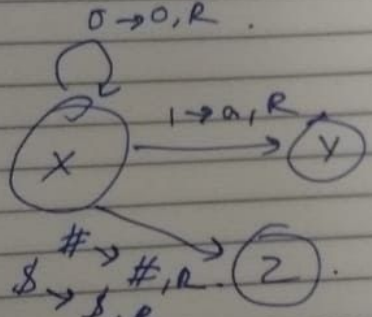


Q 2.

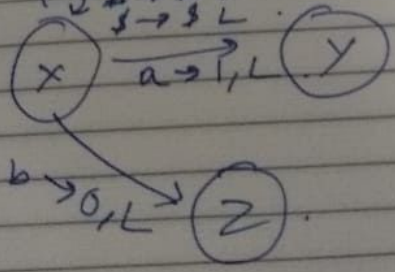
Next row.



Find 1



Reset. 1 → 1, R.
0 → 0, R.



Never give up, for that is just the place and time that the tide will turn.