Discrete Structure & Algorithms

16 / 09 / 2016   Math

60% Final ⟶ { Theory
                Problem Solving

50%   50%
Data Structure   Math in Logic
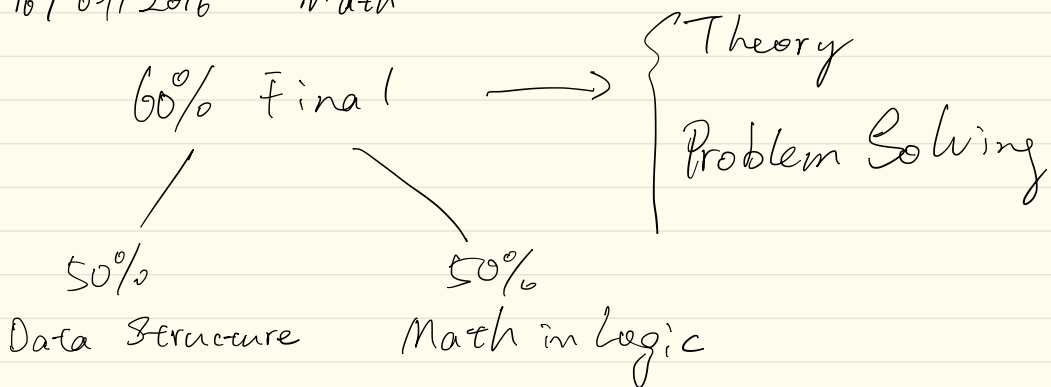
Logic   — 34%
Graphics   — 33%
Cytography — 33%

¬(A∨B) ⟺ ¬A ∧ ¬B   ¬(A∧B) ⟺ ¬A ∪ ¬B

A	B	A∨B	¬(A∨B)	¬A	¬B	¬A∧¬B
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

A	B	¬A	¬B	A∧B	¬(A∧B)	¬A∨¬B
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

3/10/2016   Linked List

Curr = head

while (curr. next != null) {

        Curr = curr. next;

    }

Traverse

        curr = head

        while (curr != = 0   ) {

                curr = curr. next

            }

Insert @ head

        Insert (L list, ē ele) {
          ele.next = head.next;
          head. next = ele

        {

is Empty
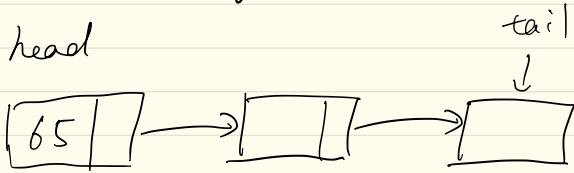
if (head == null)
        return true
return false

QueueArray

```
int[] array = new int[100];
int count = 0;

void inqueue(int num){
    array[count] = num;
    count ++;
}

void dequeue(){
    move each element i cell back
```

Implement Queue using Linear list

head

tail

```
[ 65 | ]  ──→  [   |  ]  ──→  [      ]
```

enqueue at tail

dequeue at head