Todays Content

1. Count Set Bits
2. Set x & y^{th} bit
3. Continous x set bits & y unset bits
4. Unset i^{th} Bit

Revision:

```
int Set(int N, int i){ TC: O(1) SC: O(1)
    N = N | (1<<i)      # Set ith bit in N
    return N;
}
```

```
boolean checkBit(int N, int i){ TC: O(1) SC: O(1)
    return (N>>i)&1 == 1;
}
```

Count Set Bits:

Given N, return no: of set bits in N.

Ex1:

	2^5	2^4	2^3	2^2	2^1	2^0
N= 21	0	1	0	1	0	1

return 3

	2^5	2^4	2^3	2^2	2^1	2^0
N= 45	1	0	1	1	0	1

return 4

Idea1: Iterate on all bits from 0..31 & if bit is set: c++;

```
int c=0;
for(int i=0; i<32; i++){   # Iterations = 32  TC: O(1)
    if((N>>i) &1 ==1){
        c++;
    }
}
return c;
```

Idea2:

	2^5	2^4	2^3	2^2	2^1	2^0	N&1	c=0
N= 45	1	0	1	1	0	1	✓	c=1 ✓
N>>1	_	1	0	1	1	0	*	
N>>1	_	_	1	0	1	1	✓	c=2 ✓
	_	_	_	1	0	1	✓	c=3 ✓
	_	_	_	_	1	0	*	
	_	_	_	_	_	1	✓	c=4
	_	_	_	_	_	0	Stop	return c=4.

```
int countSet(int N){
    int c=0;
    while(N>0){
        if(N&1 ==1){
            c++;
        }
        N= N>>1;
    }
    return c;
}
```

Issue:

```
N=10;
print(N+10) ; 20
print(N) ; 10
```

```
N=10;
print(N>>1) 5
print(N) 10
```

We need to use assignment operators to update value N.

3^{rd} Approach:

N								N-1								N & (N-1)						
	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^6	2^5	2^4	2^3	2^2	2^1	2^0
N=49	0	1	1	0	0	0	$\cancel{1}$	N=48	0	1	1	0	0	0	0	0	1	1	0	0	0	0
N=24	0	0	1	$\cancel{1}$	0	0	0	N=23	0	0	1	0	1	1	1	0	0	1	0	0	0	0
N=8	0	0	0	$\cancel{1}$	0	0	0	N=7	0	0	0	0	1	1	1	0	0	0	0	0	0	0
N=52	0	1	1	0	$\cancel{1}$	0	0	N=51	0	1	1	0	0	1	1	0	1	1	0	0	0	0
N=42	0	1	0	1	0	$\cancel{1}$	0	N=41	0	1	0	1	0	0	1	0	1	0	1	0	0	0

#obs: N&(N-1) : Will unset right most set bit.

```
int c=0;   # iteration = No: of set bits
while(N>0) {
    N = N&(N-1);  # In each iteration right most bit becomes unset
    c++;
}
return c;
```

	2^5	2^4	2^3	2^2	2^1	2^0	
N=45	1	0	1	1	0	1	
N=N&(N-1)	1	0	1	1	0	0	c++;
N=N&(N-1)	1	0	1	0	0	0	c++;
N=N&(N-1)	1	0	0	0	0	0	c++;
N=N&(N-1)	0	0	0	0	0	0	c++;

N==0 : return c = 4;

2Q Given x & y : Set x^{th} & y^{th} Bit in 0.

Constraints

$0 <= x, y <= 30$

Ex1:

$x = 3$ $y = 5$ ans = 40

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	0	0	0

Ex2:

$x = 2$ $y = 4$ ans = 20

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	0	1	0	0

Ex3:

$x = 3$ $y = 1$ ans = 10

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	1	0

```
int setBits(int x, int y){

    return (1<<x) | (1<<y)   ✓
    return (1<<x) ^ (1<<y)   # if x==y we get zero
}
```

30 Given x, y set consecutive x bits & y unset bits.

Ex:

$x=3$ $y=2$ ans $=28$ | $x=4$ $y=3$ ans $=120$ | $x=2$ $y=5$ ans $=96$

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	1	1	0	0
1	1	1	1	0	0	0
1	1	0	0	0	0	0

Constraints $0 <= x+y <= 30$

Ideas

1. Solving with loops : TODO
2. Without loops

Hint: $2^0 + 2^1 + 2^2 + \dots 2^N = 2^{N+1} - 1$; # $2^0 + 2^1 + 2^2 + 2^3 = 2^4 - 1 \Rightarrow 15$

Ex:

1. $x=3$ $y=2$

2^6	2^5	2^4	2^3	2^2	2^1	2^0
				1	1	1
		1	1	1	0	0

$= 2^0 + 2^1 + 2^2 = 2^3 - 1$

$(2^3-1) << 2$

2. $x=4$ $y=3$

2^6	2^5	2^4	2^3	2^2	2^1	2^0
			1	1	1	1
1	1	1	1	0	0	0

$= 2^0 + 2^1 + 2^2 + 2^3 = 2^4 - 1$

$(2^4-1) << 3$

3. $x=2$ $y=5$

2^6	2^5	2^4	2^3	2^2	2^1	2^0
					1	1
1	1	0	0	0	0	0

$= 2^0 + 2^1 = 2^2 - 1$

$(2^2-1) << 5$

$a << n = a * 2^n$

Given x, y: $\underbrace{(2^x - 1)}_{a} << y \rightarrow \boxed{(1 << x - 1) << y}$

$a << y \rightarrow a * 2^y = (2^x - 1) * 2^y = 2^{x+y} - 2^y =$

$\boxed{= 1 << (x+y) - 1 << (y)}$

Q8: Given N & i : # If i^{th} Bit in N is already Unset : Leave it
if i^{th} Bit in N is Set : Unset

Ex1:

N=45 i=3

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Unset i=3	0	1	0	1	1	0	1
ans = 37	0	1	0	0	1	0	1

N=57 i=3

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Unset i=3	0	1	1	1	0	0	1
ans = 49	0	1	1	0	0	0	1

Idea: 1 & 0 = 0   0 & 0 = 0

N=45 i=3

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Unset i=3	0	1	0	1	1	0	1
(1<<3)	0	0	0	1	0	0	0
~(1<<3)	1	1	1	0	1	1	1
N & ~(1<<3)	0	1	0	0	1	0	1

N=57 i=4

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Unset i=3	0	1	1	1	0	0	1
(1<<4)	0	0	1	0	0	0	0
~(1<<4)	1	1	0	1	1	1	1
N & ~(1<<4)	0	1	0	1	0	0	1

```
int Unset(int N, int i){

    N = N & ~(1<<i)
    return N;
}
```