Lecture 10: Bit Manipulation - 2

Represent -3 in binary

+3 : 0 0 0 0 0 0 1 1

2's complement 1 1 1 1 1 1 0 0

\+ 1

1 1 1 1 1 1 0 1 ⇒ -3

(28) Help from Sam.

If Alex solves a question → score doubles
takes help → score +1

Initial score : 0 Target : 5

0
help 1
solves 2
solves 4
help 5

Ans. (2) help taken

```
if A == 0:
    return 0
cnt = 0
while A:
    cnt += 1
    A = A & (A-1)
return cnt
```

In hindsight of the story, this problem basically is checking the ON bits

(29) Given an array of integers A. Two integers appear only once & all others twice. Find them.

[1 2 3 1 2 4] ⇒ [3, 4]

```
xor = 0
answer = [0, 0]

for i in range(n):
    xor ^= A[i]

msb = 0
while xor:
    xor = xor & (xor-1)
    msb += 1

set1 = []
set2 = []

for e in A:
    if e >> (msb-1) & 1:
        set1.append(e)
    else:
        set2.append(e)

for x in set1:
    ans[0] ^= x
for y in set2:
    ans[1] ^= y
return sorted(ans)
```

xor == 7 [0 1 1]

finding most significant bit

even no. set
odd no. set

(30) Maximum Satisfaction

Given an array A on N integers. A[i] represents quality of the i^{th} fruit. Pick four fruits s.t. satisfaction value will maximum. → a & b & c & d

(& ↑ bitwise AND)

Find maximum satisfaction value.

This means we have to check which MSB is ON using the integer's 32 bit binary rep"

```
ans = 0
for i in range(32, -1, -1):
    temp = ans | (1<<i)
    if check(temp, A) == 1:
        ans = temp
return ans
```

```
def check(x, A):
    cnt = 0
    for a in A:
        if (a & x) == x:
            cnt += 1
        if cnt == 4
            return 1
    return 0
```

(31) Excel column number

"AB" ⇒ 28 ((1)(26¹) + (2)(26⁰))

"YZ" ⇒ 25(26) + (26)(1)

```
N = len(A)
ans = 0
for i in range(N-1, -1, -1):
    ans += (ord(A[i]) - 64) * (26 ** (N-1-i))

return ans
```