nim magic formula

consider t-pile nim position $p = (p_1, p_2, \ldots, p_t)$

definition nim-sum(p): xor(p) $= p_1 \oplus p_2 \oplus \ldots \oplus p_t$

theorem (Bouton, 1901): player-to-move wins (has winning strategy)

if and only if nim-sum(p) $\neq 0$

pile	stones	stones (binary)
a	1	1
b	2	1 0
c	6	1 1 0
nim_sum(1 2 6)	5	1 0 1

all winning moves from nim(1 2 6) ?   try all moves

(0 2 6) (1 1 6) (1 0 6) (1 2 5) (1 2 4) (1 2 3) (1 2 2) (1 2 1) (1 2 0)

```
0         0     1         1     1         1     1         1     1         1

2       1 0     1         1     0         0     2       1 0     2       1 0

6   1 1 0       6   1 1 0       6   1 1 0       5   1 0 1       4   1 0 0

---------       ---------       ---------       ---------       ---------

    1 0 0           1 1 0           1 1 1           1 1 0           1 1 1
```

```
1         1     1         1     1         1     1         1

2       1 0     2       1 0     2       1 0     2       1 0

3       1 1     2       1 0     1         1     0         0

---------       ---------       ---------       ---------

          0               1             1 0             1 1
```

all nim(1 2 6) winning moves?

change nim-sum from xor(1 2 6) = 5   to 0

change nimsum by 5?   change k-pile to (k xor 5)-pile

pile a? change 1-pile to (1 xor 5 = 4)-pile ? not legal nim-move

pile b? change 2-pile to (2 xor 5 = 7)-pile ? not legal nim-move

pile c? change 6-pile to (6 xor 5 = 3)-pile ? yes !

how to find all winning moves:

* find leftmost column with nimsum bit 1
* each pile with 1 in that column has winning move

pile	stones	stones (binary)
a	1	1
b	2	1 0
c	6	1 1 0
nim_sum(1 2 6)	5	1 0 1
leftmost column		*

piles with 1 in column *? a no   b no   c yes

only win-move: pile c, 6-pile => (6 xor 5 = 3)-pile:

take 3 stones, leave 3 :)

another example

a	1 0 0	4 stones
b	1 0 1	5 stones
c	1 1 1	7 stones
	------	
	1 1 0	

find all winning moves

pile a

```
      1 0    (take 2 stones) leave 2
    1 0 1        5 stones
    1 1 1        7 stones
   ------
        0
```

pile b

```
    1 0 0        4 stones
      1 1     (take 2 stones) leave 3
    1 1 1        7 stones
   ------
      j 0
```

pile c

```
 1 0 0        4 stones
 1 0 1        5 stones
     1      (take 6 stones) leave 1
------
     0
```

another example: find all winning moves

```
1 0 1 1 0
  1 1 0 1
1 1 0 1 1
    1 0 1
---------
    1 0 1
```

check your answer with class program nim/nimbig.py