LC 136

class Solution:

def singleNumber(self, nums: List[int]) -> int:

res = nums[0]

for num in nums[1:]:

res ^= num

return res

LC 201

class Solution:

def rangeBitwiseAnd(self, left: int, right: int) -> int:

a = format(left, 'b')

b = format(right, 'b')

if len(a) != len(b) or left == 0:

return 0

res = 0

cur = 2 \*\* (len(a)-1)

for i in range(len(a)):

if a[i] == b[i]:

if a[i] == '1':

res += cur

cur = cur // 2

else:

break

return res

LC 318

class Solution:

def maxProduct(self, words: List[str]) -> int:

nums = []

lengths = []

for word in words:

num = 0

for j in set(word):

num += 2 \*\* (ord(j)-97)

nums.append(num)

lengths.append(len(word))

res = 0

for i in range(len(nums)-1):

for j in range(i,len(nums)):

if nums[i] ^ nums[j] == nums[i] + nums[j] and \

nums[i] | nums[j] == nums[i] + nums[j]:

res = max(res,lengths[i]\*lengths[j])

return res

LC 371

class Solution:

def getSum(self, a: int, b: int) -> int:

x,y = abs(a),abs(b)

if x<y:

return self.getSum(b,a)

if a<=0:

sgn = -1

else:

sgn = 1

c = format(x,'010b')

d = format(y,'010b')

if a\*b<0:

subtract = '0'

res = ''

for i in range(9,-1,-1):

if subtract == d[i]:

res = "".join((c[i], res))

elif c[i] == '1':

res = "".join(('0', res))

subtract = '0'

else:

res = "".join(('1', res))

subtract = '1'

else:

add = '0'

res = ''

for i in range(9,-1,-1):

if c[i] == d[i]:

res = "".join((add, res))

if c[i] == "1":

add = '1'

else:

add = '0'

elif add == '1':

res = "".join(('0', res))

else:

res = "".join(('1', res))

if add == '1':

res = "".join(('1', res))

return sgn \* int(res,2)