

Q ^{google} You have two 1D arrays of length n , where n is odd.

We need to check, if we can form pair of elements _(p_1, p_2) from both the arrays such that xor of each pair

is same.

$n \leq 10^6$

$[1, 1, 1]$ $[2, 2, 2]$

$1 \wedge 2 =$

$1 \wedge 2 =$

$1 \wedge 2 =$

$[a, b, c]$ $[d, e, f]$

| |
|------------------|
| $a \wedge e = v$ |
| $b \wedge d = v$ |
| $c \wedge f = v$ |

\rightarrow 204

↳ 1) → think w.r.t no properties

$$A = [a, b, c] \quad B = [d, e, f]$$

we get all pairs \rightarrow $\boxed{nC_2}$ $n \leq 10^6$ ~~Bruteforce~~

we will
eventually
get some
combinations

$$\leftarrow \boxed{a \wedge f} = \boxed{b \wedge d} = \boxed{c \wedge e} = \boxed{V} \rightarrow \boxed{\begin{matrix} \text{yes} \\ \text{no} \end{matrix}}$$

A number $\rightarrow x$

;

$$\begin{aligned} x \wedge x &= 0 \\ x \wedge 0 &= x \end{aligned}$$

odd length — $\boxed{a \wedge f} = \boxed{b \wedge d} = \boxed{c \wedge c} = \underline{\underline{v}}$

$\boxed{a \wedge f} \wedge \boxed{b \wedge d} \wedge \boxed{c \wedge c}$

if we xor all the element

~~1~~ $a \wedge b \wedge c \wedge d \wedge c \wedge f = c \wedge c = \underline{\underline{v}}$

$$a \wedge b \wedge d \wedge f = 0$$

$$a \wedge b \wedge d \wedge f = \underline{a \wedge f \wedge b \wedge d}$$

$$\boxed{a \wedge b \wedge c} = \boxed{c \wedge b \wedge a} = \boxed{c \wedge a \wedge b}$$

$$\begin{aligned} x \wedge x &= 0 \\ 0 \wedge x &= x \end{aligned}$$

$$\underline{\underline{x \wedge x \wedge x = x}}$$

if n is odd

odd pairs

$\odot v \rightarrow$ XORing all the elements from both arrays.

$\pi(O(n))$
 $\delta(O(n))$

$$a \wedge f = v$$

$$b \wedge d = v$$

$$c \wedge e = v$$

$\{a, b, c\}$

$\{d, e, f\}$

hashmap

$O(1)$

$$a \wedge f = v$$

then XORing f on both sides

$$a \wedge f \wedge f = v \wedge f$$

$$a \wedge 0 = v \wedge f$$

$$\boxed{a = v \wedge f}$$

$$\boxed{v = a \wedge f}$$

$$b = v \wedge d$$
$$d = v \wedge b$$

IP \rightarrow IPv4

Q. Given a string, containing only digits, you have to return all possible IP addresses that can be made from the string.

from the string

$S = "25525511135"$

$\hookrightarrow "255.255.111.35"$
 $\hookrightarrow "255.255.11.135"$

idx $\rightarrow 0$

$0 + 1$

$0 + 1$

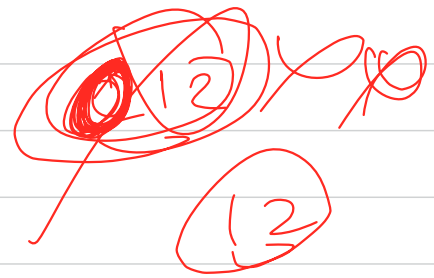
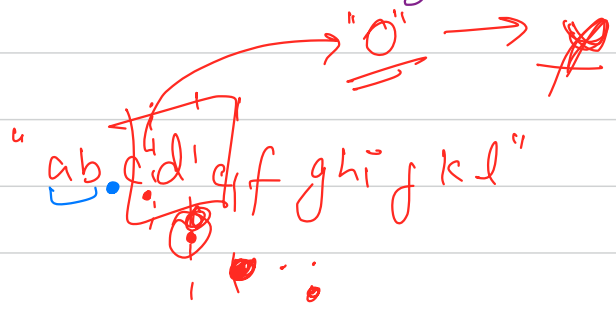
$0 + 1$

$2 + 1$
 $2 + 1$
 $2 + 1$

$S.size \leq 3 \times 10^3$

$> 12 \text{ bytes}$

for 1Pv₁ → 4 intger [0-255]

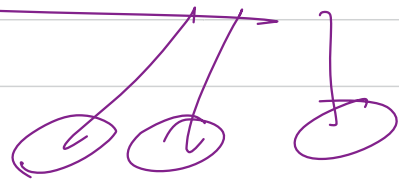


count of dots

Reverse

count of dots

and for the next 3 position



- 3)
- Base Case
 - Recur task
 - Self work

Base Case

If the index $> 4 \rightarrow$ return γ

(if the index $= 4$ and $i == \text{str.size}()$)

print (output)

Self work \rightarrow to extract the next possible char

~~start $= 0$ $\text{str.size}()$~~ ~~loop body \rightarrow dots~~ ~~$i = i + 1$~~

recursive \rightarrow to place dots for rest of the string

2.3.4.5.6.7.8.9.10.11.12.13.14.15.16.17.18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.36.37.38.39.40.41.42.43.44.45.46.47.48.49.50.51.52.53.54.55.56.57.58.59.60.61.62.63.64.65.66.67.68.69.70.71.72.73.74.75.76.77.78.79.80.81.82.83.84.85.86.87.88.89.90.91.92.93.94.95.96.97.98.99.100.

0

21

22.0.0

22.0.0

21

Q2 "hi" \rightarrow hiiii ; "hello" \rightarrow heellllooo
original stretch word of same original len

You are given a string S , & a list of strings.
You need to check for how many strings in the list

the given string S is a stretchy string. How can we
make a normal string stretchy \rightarrow choose a group consist
of C characters, & add some replica of C to the
group so that size of group is 3 or more.

"hell" \rightarrow hello \rightarrow hellllloo llloo
llloo
23

$S = \text{hellloo}$

$l = [\text{"helo", "hi", "hello"}]$

\rightarrow 1

$S.length \leq 100$

$l.size() \leq 100$

$l[i].length \leq 100$

if $l[P_i] == S$

→ Yes ~~or~~ ~~can~~

if $l[P_i]$ contains a char which is not present in S

or vice-versa → NO

2
"hello"
→ ll
→ C_1

"hellloo"
→ ll
→ C_2

consecutive
same char
→ $C_1 == C_2$

$C_1 \Rightarrow C_2$
 $C_2 \Leftarrow 3$
→ $C_1 == C_2$
→ $C_1 \neq C_2$