① Use two loops and check, whether element is unique or not.

→ if not unique : push into vector.

② Using sorting

→ Check adjacent elements.

$a[i] == a[i+1]$:→ push into vector

③ Using frequency array → range of values not more than 10^5, 10^6

↳ we can use frequency array : $freq[i] > 1$ ⇒ push

a =	4	3	2	7	8	2	3	1
indx	0	1	2	3	4	5	6	7

n-different elements → and you need track how many times they occur.??

Values : 1 → n

they are not negative and can be represented with indices i+1 → repeat

4	3	-2	-7	8	2	3	1
0	1	2	3	4	5	6	7

$1-1=0$ F

$2-1=1$ F

$3-1=2$ F

4 → $4-1=3$ F

$5-1=4$ F

$6-1=5$ F

$7=6$ F

$8-1=7$ F

$(1 \to n)-1$

$(0 \to n-1)$

```
for (i=0; i<n; i++){
    check whether visited → if (a[i] < 0) {
    mark it visited
    a[a[i] - 1] *= -1;
}
```

- initially all elements are in 1 to n

- for an element a[i] we go the index (a[i] - 1)

and negate the value (change its sign) ⇒ +ve → (−) ve

(−) ve → (+) ve

⇓

it doesnot affect the original value there,

[1, 3] - - - - - - .. >3

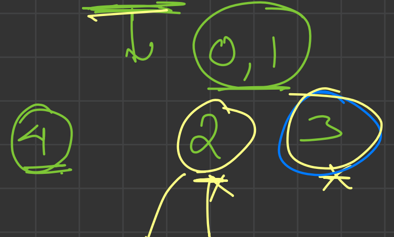
n = 6

(1 → n)

maxSumPossible = 0

patches = 0 =

[1, 4]



i = 0; a[i] = 1 ⇒ maxSumPossible = 1

$i = 1$: $a[i] = 3$ → we can't make $\boxed{2}$

$\boxed{3}$

0, 1, i need a patch : 1 ✓

2 ✓

1 $\quad$ $\boxed{2}$

(1, 1, 4]

↓

1, 2, 4 _ _

(1, 2, 4)

↓

1, 2, 3, 4

$\boxed{0, 1, 2, \dots, x}$ $\boxed{x+1}$, $x+2$, $x+3$ . . . $\boxed{2x+1}$ $\cancel{3}$

$\boxed{x+1}$

⇓

guarantee

$\boxed{2} + \boxed{x-1}$

⇓

$\boxed{x+1}$ $\quad$ $\boxed{x+1}$ $\quad$ $\boxed{2x+1}$

$\boxed{8}$ $\boxed{8}$ $\boxed{8}$ $\quad$ $\underset{x}{\boxed{8}}$

Sums: 1, 2, 3, 4, 5, 6, 7, $\cancel{8}$

$1 \longrightarrow \boxed{7}$ $\quad$ $\boxed{8} \rightarrow 8$ $(1+8) \rightarrow (7+8)$

$\boxed{1 \rightarrow 8}$ $\boxed{15}$ ✓

$\{[1, 2, 4, 8]\}$

$+$

2.

3

$1+2+4+1 = 8$

$2+4+2 = 8$

$4+1+3 = 8$

$\vdots$

$1+7 = 8$

$8 = 8$

i can't pick 9

an element $> x+1$

patch

$1 \rightarrow$ range: $1 \rightarrow 8$

$2 \rightarrow$ range: $1 \rightarrow 9$

$3 \rightarrow$ range: $1 \rightarrow 10$

$\vdots$

$8 \rightarrow$ range: $1 \rightarrow 15$: $x \mathrel{+}= (x+1)$

$[1, 2, 3]$ $n = 6$ ~~1~~ 2 3 4 5 6

$\boxed{x = 3}$

① → using elements till 1) can change
x to 1

$0, \ldots x$ $(x+1)$

③ → $0, 1, \boxed{2}$ ③

$a[i] > x+1$ → $a[i]$ → should be anything in 1 to $x+1$

$3 > 1+1$

⇒ we need a patch: we use patch of $x+1$ (2)

$x \mathrel{+}= \underline{(x+1)};$

$x \mathrel{+}= (2); \Rightarrow \underline{\underline{x = 3}}$

$a[i] > x+1$ which means we don't need a patch at this moment.

$3 > 3+1 = \boxed{4}$

$\boxed{3}$ $\boxed{x = 3}$ $x = 0, \underline{1, 2}, 3$ $\boxed{3}$

$\downarrow$

$\boxed{1, 2}$

We update $\boxed{x}$ as $x \mathrel{+}= \underline{a[i]};$

all sum possible from 1 to x

1 to 4

$\boxed{maxlength = 3}$

a	b	c	a	b	c	b	b
0	1	2	3	4	5	6	7
						j	i

$\underline{(i - j + 1)}$

$7 - 7 + 1 = 1$

freq

a →	1
b →	1
c →	1

Divisible Subarrays → HB

Remove all adjacent duplicates in a string → LC (1047)

1209