

Vowels in Range

0 1 2 3 4 5

A = scalex.

$$Q = [0 \ 4] \rightarrow 2$$

$$[2 \ 4] \rightarrow 2$$

$$[0 \ 2] \rightarrow 1$$

s c a l e x .

0 0 1 0 1 0

$$P[i] = 0 \ 0 \ 1 \ 1 \ 2 \ 2$$

vowels = a, e, i, o, u.

Minimum Pidu.

$$A = [5 \ 17 \ 100 \ 1]$$

$$\begin{aligned} \text{ans} &= [\text{max}[\text{of Even}] - \text{min}[\text{odd}]] \\ &= [100 - 1] \\ &= 99. \end{aligned}$$

even \rightarrow if (arr[i] % 2 == 0)

{ else { // odd
|
}

$$A = [0 \ 2 \ 9]$$

$$\begin{aligned} &[2 - 9] \\ &= -7 \end{aligned}$$

little pony & Bitwise Operator - 2

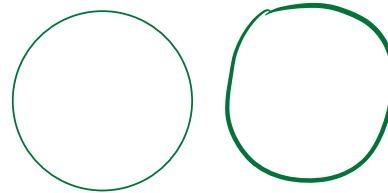
Disjoint Subarray \rightarrow

\downarrow
no overlap.



No

Bitwise AND of 1st subarray
+
Bitwise OR of 2nd subarray
= maximum.



$$A = [5 \ 5 \ 4 \ 3 \ 4]$$

AND $\rightarrow 5$

OR $\rightarrow 7$

[0 0] $\rightarrow 5$

[1 0]

[0 1] $\rightarrow 5$

[1 2]

[0 2] $\rightarrow 4$

[1 3]

[0 3] $\rightarrow 0$

[0 4] $\rightarrow 0$

$$4: 0100$$

$$3: 0011$$

$$\hline 0000$$

$$4: 0100$$

$$5: 0101$$

$$\hline 0100$$

$$\text{Sum} = 5 + 7 = 12$$

$$A = [3 \ 5 \ 6 \ 1]$$

$$3: 0011$$

$$5: 0101$$

$$\hline \text{OR} : 0111$$

AND = 6

OR = 7

$$\text{Sum} = 7 + 6 = 13$$

AND \rightarrow cannot increase
 \downarrow same or decrease

N: x x x x x

A: y y y y y

$$\begin{array}{r} 10101 \\ 11111 \\ \hline 10101 \end{array}$$

Obs: AND of a number can remain same or decrease

OR: cannot decrease
same or increase.

OR:
$$\begin{array}{r} 10101 \\ 00000 \\ \hline \text{OR: } 10101 \end{array}$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ [2 & 32 & 8] \end{array}$$

prefixed
i-1

AND = 32

$$\begin{array}{r} 10101 \\ 10101 \\ \hline 10101 \end{array}$$

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 1 & 2 & 1 \\ \text{pfor} = & 1 & 1 & 3 & 3 \\ \text{sfor} = & 3 & 3 & 3 & 1 \end{array}$$

i+1 suffix 0 n

op1: OR = 2

op2: OR = (8 | 4) = 12

sum = 34.

sum = 44.

AND = 4

OR = [2 | 32 | 8] = 42

sum = 46

$O(N) \neq N$
TC: $O(N^2)$

$N = 10^5$
 $N^2 = 10^{10} \rightarrow \text{TLE}$
100%.

$$\begin{array}{cccc} \text{index} & 0 & 1 & 2 & 3 \\ \text{arr} & [1 & 1 & 2 & 1] \\ \text{pfor} & 1 & 1 & 3 & 3 \\ \text{sfor} & 3 & 3 & 3 & 1 \end{array}$$

// pfor $\rightarrow O(N)$
// sfor $\rightarrow O(N)$
} TO DO

for (i = 0; i < N; i++) {
tempAns = max(pf[i-1], sf[i+1]) + arr[i]
ans = max(ans, tempAns)
}

TC: $O(N+N+N) = O(3N) = O(N)$

SC: $O(2N) = O(N)$

$N = 10^5$

7: 22. $\rightarrow 11$
 8: 52 $\rightarrow 157$.

Removing Subarray.

$$A = [[7] \ 3 \ 1]$$

$\underbrace{\quad\quad}_4$ $\underbrace{\quad}_2$
 $\rightarrow \text{len} = 1.$

$$B = 3$$

$$A = [\begin{matrix} \downarrow & \downarrow \\ 5 & 10 & 15 & 20 \end{matrix}]$$

$\underbrace{\quad\quad}_5$ $\underbrace{\quad\quad}_5$ $\underbrace{\quad}_5$

$> 10^8$ TLE

$$B = 4.$$

$$N = 10^3.$$

$$N^2 = 10^9$$

BF: $N^2 \cdot N = O(N^3)$

$\nearrow \frac{N(N+1)}{2}$

ex: $[\underbrace{2 \ 4 \ 1 \ 3 \ 1}_{2 \ 2 \ 2 \ 2} \ 8 \ 6 \ \underbrace{2 \ 4 \ 1 \ 3 \ 1}_{2 \ 3 \ 2 \ 2}]$ $B = 5$

$\downarrow s_i$ $\downarrow e_i$

$$\frac{N}{2} \times \frac{N}{2} = N^2$$

$$A = [8 \ \underbrace{2 \ 4 \ 1 \ 4}_{6 \ 2 \ 3 \ 3}]$$

$$B = 5.$$

```

flag = false
for (i = 0; i < N; i++) {
    if (abs(a[i] - a[i-1]) > B) {
        flag = true; break;
    }
}
if (!flag) return 0;

```

```

Si = 0
for (i = 0; i < N; i++) {
    if (abs(arr[i] - arr[i-1]) <= B)
        Si = i;
    else
        break;
}

```

```

ei = N-1
for (i = N-2; i >= Si; i--) {
    if (abs(arr[i] - arr[i+1]) <= B) {
        ei = i;
    } else
        break;
}

```

ans = N;

```

for (l = Si; l >= 0; l--) {
    for (r = ei; r < N; r++) {
        // remove [Si+1 ei-1] = ei
        len = r - l - 1;
        if (abs(arr[l] - arr[r]) <= B) {
            ans = min(ans, len); break;
        }
    }
}

```

$$[a, b] = b - a + 1.$$

$$ei - 1 - [Si + 1] + 1$$

$$= ei - 1 - Si - 1 + 1$$

$$= ei - Si - 1$$

$$\text{// remove } [Si+1, ei-1] = ei$$

$$len = r - l - 1$$

$$\text{if } abs(arr[l] - arr[r]) \leq B \text{ then } ans = \min(ans, len); \text{ break;}$$

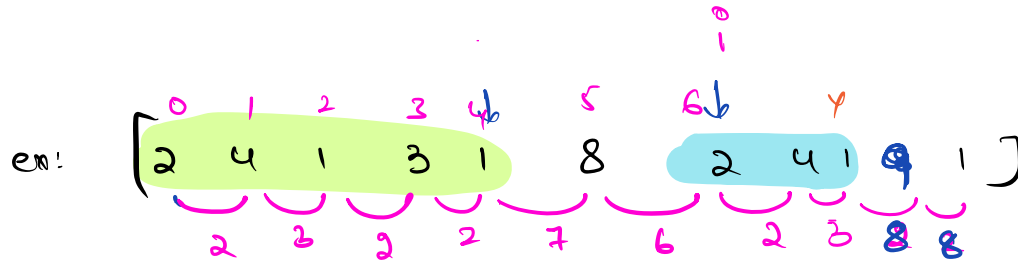
```

    }
    return

```

$N = 10^3$
 $Tc: O(N^2) \rightarrow 10^6$

Small Sub = 0



smallest Sublen = 0

$i = 0$

for $i = 0 ; i < N - 1 ; i++$

if $(\text{abs}(A[i] - A[i+1]) > B)$

smallestSublen $++$;

else if (smallestSublen > 0)

break;

}