1. Largest Palindrome Substring
2. Alternate Subarray.
3. Valid Sudoku.

Q.1 Given a string, find the length of largest palindromic substring.

Amazon
GS | MS
Directi
. . .

3

S: abacab ⇒ 5

S: abcdb ⇒ 1

Quiz S: abaeabf

5

Brute Force

→ Iterate over all the substrings ⇒ $O(N^2)$

→ Check if a substring is palindrome or not.
→ $O(N)$

TC : $O(N^3)$
SC : $O(1)$

Quiz

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x	b	d	y	z	z	y	d	b	d	y	z	y	d	x
i 0									j 9					

```
int lengthPalindrome(S, Ci, Cj)
    i = Ci, j = Cj;
    while ( i >= 0 && j < N) {
        if (S[i] == S[j])
            i--, j++;
        else
            break;
    }
    return j-i-1;
}
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
x	b	d	y	z	z	y	d	b	d	y	z	y	d	x
i														

i=0 → fun(S, 0, 0) ⇒ 1-(-1)-1 = $\boxed{1}$ ✓

→ fun(S, $\underset{i}{0}$, $\underset{j}{1}$) ⇒ 1-0-1 = $\underline{0}$

```
ans = 1;
for (i = 0; i < N; i++){
    // Odd length palindrome
    ans = max(ans, lengthPalindrome(s, i, i));
    // Even length palindrome
    ans = max(ans, lengthPalindrome(s, i, i+1));
}
return ans;
```

TC : $O(N^2)$

SC : $O(1)$

$$\begin{array}{c} TC : O(N^2) \\ SC : O(1) \end{array} \longrightarrow \begin{array}{c} TC : O(N^2) \\ \underbrace{SC : O(N^2)}_{\text{DP}} \end{array} \longrightarrow \begin{array}{c} TC : O(N) \\ \text{Manacher's Algo} \end{array}$$

Sol^n

Q. Valid Sudoku

Given a partially filled sudoku, check if it's valid or invalid.

9x9

	0	1	2	3	4	5	6	7	8
0	1	8	4						6
1				9		7		3	
2					6				
3	8			5			6		
4		1				4			
5			2			3			
6									
7			1		2			8	
8		7					1		

Rules :

1) No repetition in any row.
2) No repetition in any col.
3) No repetition in any 3x3 box

```
// Rows
for ( i= 0; i< 9 ;  i++ ){
    HashSet<int> set;
    for ( j = 0; j<9 ; j++ ) {
        if ( set.contains ( mat[i][j] ) )
            return false ;
        else
            set.add (mat[i][j]);
    }
}
// Cols
for ( i= 0; i< 9 ;  i++ ){
    HashSet<int> set;
    for ( j = 0; j<9 ; j++ ) {
        if ( set.contains ( mat[j][i] ) )
            return false ;
        else
            set.add (mat[j][i]);
    }
}
// 3×3  Bones
→  (0,0)   (0,3)   (0,6)
→  (3,0)   (3, 3)   (3,6)
→  (6,0)   (6,3)   (6,6)
```

```
for( i=0; i<9 ; i+=3) {
    for( j=0; j<9 ; j+=3) {
        // i,j ⇒ Start index of 3x3 Box.
        HashSet<int> set;
        for( k=i ; k< i+3; k++){
            for( l=j; l<j+3; l++){
                if( set.contains( mat[k][l]))
                        return false;
                else
                        set.add(mat[k][l]);
            }
        }
    }
}
```

# of iterations $= 3 \times (9 \times 9)$

$= 3 \times 81$

$= \underline{\underline{243.}} \approx \underline{\underline{O(1)}}$ $\underline{\underline{TC}}$

$\underline{\underline{N=9}}$

# of iterations $= 3 \times N^2$

TC : $\underline{\underline{O(N^2)}}$

Q. Alternate Subarrays.

Given an int array A of size N comprising of only 0's & 1's & an integer B. Find all indices in array A that can act as a centre of a 2*B+1 length 0-1 alternate subarray.

A: [1 0 0 1 0 1 0 1 0 0 0 1 0]

{1, 0, 1, 0 - - - - } ✓ {0} ✓ {1 0 1 0 1} ✓

{0, 1, 0, 1, - - - - } ✓ {1} ✓

Ex

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [1 & 0 & 1 & 0 & 1] \end{array}$$

B = 1

2*1+1 = 3

ans: [1, 2, 3]

Ex

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ [0 & 0 & 0 & 1 & 1 & 0 & 1] \end{array}$$

B = 0

$\boxed{1}$

ans:- [0 1 2 3 4 5 6]

```
bool isAlternate(arr[], s, e){
        for(i= s+1; i<=e ; i++ ){
                if (A[i] == A[i-1])
                        return false;
        }
        return true;
}

k= 2B+1
s=0
e = k-1
while ( e < N) {
        // s,e
        // Check if subarray [s,e] is alternate.
        if( isAlternate (arr, s,e) ){
                // Add mid index in ans array.
                ans.add(s+B);
        }
        s++
        e++
}
```

$$\boxed{\begin{array}{l} TC : O(N^2) \\ SC : O(1) \end{array}}$$

* 10^8 iterations / $\underline{\underline{\text{sec}}}$

$\underline{\underline{\text{Time}}}$ $\underline{\underline{\text{limit}}}$: 1 $\underline{\underline{\text{sec}}}$

$N = 10^6 \Rightarrow O(N^2) \simeq \underline{\underline{10^{12}}} > 1\text{sec}$

$\Rightarrow \underline{\underline{\text{TLE}}}.$

$N = 10^3 \Rightarrow O(N^2) \Rightarrow \underline{\underline{10^6}} < 1\text{sec}$

$\Rightarrow$ ✓

$N = 20 \Rightarrow O(2^N)$ ✓

$\simeq 2^{20}$

$\underbrace{2^{10}}_{\underbrace{1024}_{\simeq 10^3}} \cdot \underbrace{2^{10}}_{10^3} \simeq \underline{\underline{10^6}} < 1\text{sec}$