41. First Missing Positive

APPROACH =>

BASICALLY HAM KYA KAR RHE HAI HAM HAR ELEMT KO USKI SAHI ZAGAH PE RAKH DENGE AND THEN SAB SARE ELEMNT APNI SAHI ZAGAH PE AAJYEGE TO START SE CHECK KARNEGE JO ELEMT JIS INDX PE KYA WO USI KA HAI AGAR NHI TO INDEX RETURN KARDO

AUR AGAR AISA KARTE KARTE LOOP KHATAM HOGYA ISKA MATLAB SARE ELEMNTS HAI TO NEXT SMALLEST KYA SARE ELEMNTS KE BAAD JO ATA HOGGA MATLAN SIZE+1 RETURN KARDO

EDGE CASE 1=>

DO NOT TAKE 0 AS A POSITIVE INTEGER

EDGE CASE 2=>

MAKE SURE TO CHECK THAT JO ELEMT KA IDEX DHUND RHE HO USKA IDX HAI BHI YA NHI MATLAB WO ELEMT 0 SE AARAY KE SIZE KE ANDAR HONA CHIYE;

EDGE CASE 3=>

SSWAP TABHI KARNA AGAR EDG 2 SAHI HAI END JO ELEMT SWAP HONE WLA HAI WO APNI SAHI INDEX PE NHI HAI;

EDGE 4=>

AGAR SWAP KARAN TO YAAD RKKHNA KI I++ MAT KARNA KYUI JO ELEMT HAMNE SWAP KIYA HAI USE BHI TO CHECK KRAN HAI

AGAR HAMNE SWAP NHI KIYA TABHI ELESE CASE ME I++ KARNA

```cpp

class Solution {
public:
    int firstMissingPositive(vector<int>& nums) {
        int n = nums.size();

        // Step 1: Move each positive integer to its correct index if possible
        for (int i = 0; i < n; ) {
            if (nums[i] > 0 && nums[i] <= n && nums[nums[i] - 1] != nums[i]) {
                swap(nums[i], nums[nums[i] - 1]);
            }else{
                i++;
            }
        }

        // Step 2: Find the first missing positive integer
        for (int i = 0; i < n; i++) {
            if (nums[i] != i + 1) {
                return i + 1;
            }
        }

        // If all positive integers from 1 to n are present, the missing one is n + 1
        return n + 1;
    }
};
```