# Smallest Positive Missing Number

| | |
|---|---|
| solved by | Senan |
| Platform | GeeksForGeeks |
| difficulty | Medium |
| tags | Sorting |
| language | C++ |
| solved on | @27/11/2024 |
| link | https://www.geeksforgeeks.org/problems/smallest-positive-missing-number-1587115621/1 |
| Completion | ☑ |

## Intuition

The problem requires finding the smallest positive integer missing from the array. A key insight is that the problem focuses on positive integers, so negative numbers and zero can be ignored. By using a set, we can leverage its property of uniqueness and ordered elements to efficiently check for the missing number.

## Approach

1. Insert all elements of the array into a set.
2. Initialize `pos` to 1, representing the smallest positive integer.
3. Iterate through the set:
    - Ignore elements less than or equal to zero since they aren't relevant.
    - Compare each element with `pos`. If they match, increment `pos`.
    - If an element doesn't match `pos`, then `pos` is the missing number, so break the loop.
4. Return `pos` as the answer.

## Complexity

### Time Complexity:

- **O(n log n)**: Creating the set from the array takes O(n log n) due to insertion.
- Iterating over the set is O(n).

### Space Complexity:

- **O(n)**: The set uses additional space to store unique elements from the array.

## Code

```
class Solution {
  public:
    int missingNumber(vector<int> &arr) {
        set<int> mySet(arr.begin(), arr.end());
        int pos = 1;
        for (auto &elem : mySet) {
            if (elem <= 0) continue;
            if (elem != pos) break;
            pos++;
        }

        return pos;
    }
};
```