

2442. Count Number of Distinct Integers After Reverse Operations

Logic Explanation (Step-by-step)

1. reverse_the_number function

```
int reverse_the_number(int num){
    int rev = 0;
    while(num != 0){
        rev = rev*10 + num%10; // last digit ko rev me add karna
        num = num / 10;      // last digit remove karna
    }
    return rev;
}
```

- Ye function ek integer ka **digit reverse** karta hai.
- Example: `num = 123`
 - Step 1: $rev = 0 \rightarrow rev = 0*10 + 3 = 3$, $num = 12$
 - Step 2: $rev = 3*10 + 2 = 32$, $num = 1$
 - Step 3: $rev = 32*10 + 1 = 321$, $num = 0 \rightarrow$ stop.

2. countDistinctIntegers function

```
unordered_set<int> s;
for(int i=0; i<nums.size(); i++){
    s.insert(nums[i]);          // original number ko set me daalna
    int rev = reverse_the_number(nums[i]); // uska reverse nikalna
}
```

```
s.insert(rev);           // reverse ko bhi set me daalna
}
return s.size();
```

- **unordered_set** automatically duplicates ko ignore karta hai.
- Har original number aur uska reverse set me store karte hain.
- Last me `s.size()` return karte hain jo total distinct numbers ka count hoga.

Dry Run

Test Case 1

```
nums = [1, 13, 10, 12, 31]
```

- Initially: $s = \{\}$
- $i=0$:
 - $\text{insert}(1) \rightarrow s = \{1\}$
 - $\text{reverse}(1) = 1 \rightarrow \text{insert}(1) \rightarrow s = \{1\}$
- $i=1$:
 - $\text{insert}(13) \rightarrow s = \{1, 13\}$
 - $\text{reverse}(13) = 31 \rightarrow \text{insert}(31) \rightarrow s = \{1, 13, 31\}$
- $i=2$:
 - $\text{insert}(10) \rightarrow s = \{1, 13, 31, 10\}$
 - $\text{reverse}(10) = 1 \rightarrow \text{already in } s$
- $i=3$:
 - $\text{insert}(12) \rightarrow s = \{1, 13, 31, 10, 12\}$
 - $\text{reverse}(12) = 21 \rightarrow \text{insert}(21) \rightarrow s = \{1, 13, 31, 10, 12, 21\}$
- $i=4$:

- `insert(31)` → already in s
- `reverse(31) = 13` → already in s

Final `s.size()` = 6 

Test Case 2

```
nums = [2, 2, 2]
```

- Initially: `s = {}`
- `i=0`:
 - `insert(2)` → `s = {2}`
 - `reverse(2) = 2` → already in s
- `i=1`: same as above → no change
- `i=2`: same as above → no change

Final `s.size()` = 1 

Notion-Ready Notes

Problem: Count Number of Distinct Integers After Reverse Operations

Problem Understanding

- Given an array of positive integers.
- For each integer, reverse its digits and add it to the array.
- Count how many distinct integers are present after this operation.

Approach

1. Use a set to store unique integers.
2. For each number:
 - Add the original number to the set.

- Reverse its digits and add the reversed number to the set.
3. The size of the set is the answer.

Reverse Function

```
```cpp
int reverse_the_number(int num){
 int rev = 0;
 while(num != 0){
 rev = rev*10 + num%10; // Append last digit
 num = num / 10; // Remove last digit
 }
 return rev;
}
```

## Main Logic

```
unordered_set<int> s;
for(int i=0; i<nums.size(); i++){
 s.insert(nums[i]); // Original
 int rev = reverse_the_number(nums[i]); // Reverse
 s.insert(rev);
}
return s.size();
```

## Dry Run Example 1

nums = [1, 13, 10, 12, 31]

Steps:

- Insert original & reverse → s = {1, 13, 31, 10, 12, 21}
- Final size = 6

## Dry Run Example 2

nums = [2, 2, 2]

- Only 2 in set
  - Final size = 1
- 

## Time Complexity

- $O(n * d)$  where  $d$  = number of digits in each number (max 6)
- Practically  $O(n)$

## Space Complexity

- $O(n)$  for the set

---

Bhai chahe to main tera liye iska **diagram wala Notion template** bana sakta hoon jisme array, set ka step-by-step growth dikhe arrows ke saath — isse visual learning easy ho jaayegi.

Banau?