Statement

↳ Given a string num representing an integer, determine whether it is a strobogrammatic number. Return TRUE if it is, otherwise FALSE.

Approach

↳ Initialize a map, dict, to store the valid mappings of digits that either stay the same or transform correctly when rotated 180°.

↳ Set two pointers

- left at the beginning

- right at the end

↳ Iterate from both ends of the string, comparing the pair of digits pointed to by the left and right pointers.

- Check whether nums[left] exists in dict, if not then return FALSE.

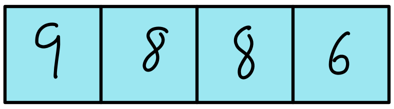
- Compare nums[left] value in dict with nums[right]. If different, return FALSE.

- Increment left by 1, decrement right by 1.

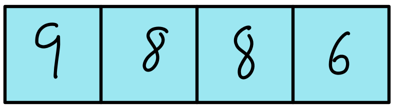
↳ Continue to iterate until left and right pointers cross.

Visualization

i) Check whether "9886" is strobogrammatic or not.

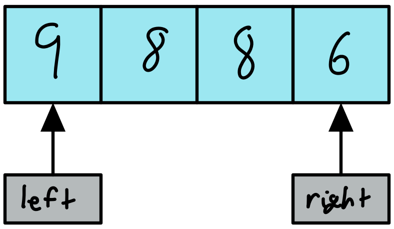


ii) Initialize a map, dict, with the digits as keys and their valid mappings.



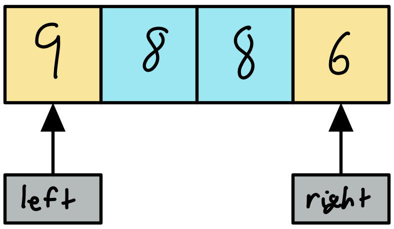
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

iii) Initialize left and right pointers



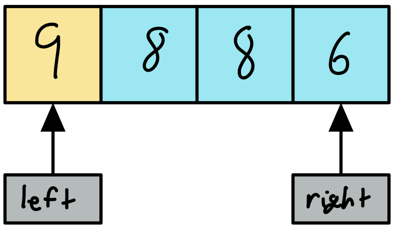
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

iv) Iterate through the string using both left and right pointers and check the pair of digits pointed by them.



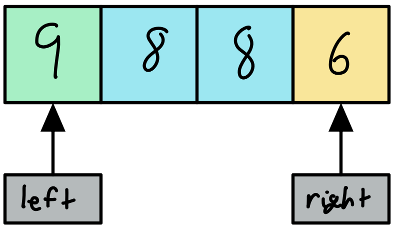
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

v) Check whether nums[left] = 9 is a valid digit that exists in dict



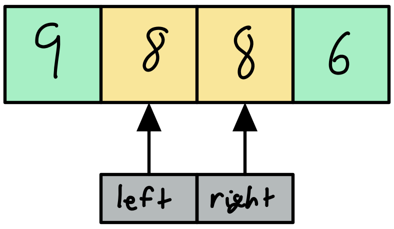
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

vi) Since nums[left]=9 exists in the dict, we retrieve its rotated value, 6, from dict and check if it matches num[right] = 6.



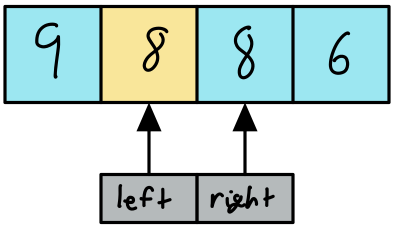
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

vii) Since the pair (9,6) is valid, move pointers inward and check next pair.



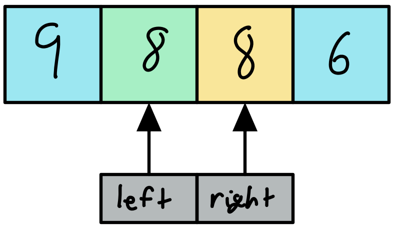
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

viii) Check whether nums[left] = 8 is a valid digit that exists in dict



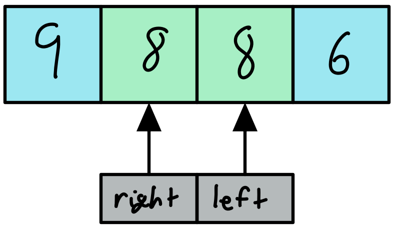
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

ix) Since nums[left] = 8 exists in the dict, we retrieve its rotated value, 8, from dict and check if it matches num[right] = 8.



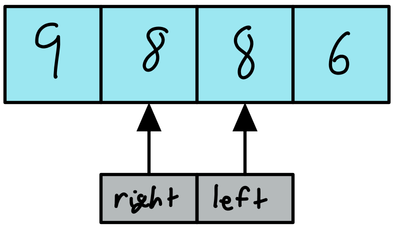
dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

x) Since the pair (8,8) is valid, move pointers inward and check next pair.



dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

xi) Stop iteration because the left and right pointer have crossed.



dict	
key	value
Ø	Ø
1	1
8	8
6	9
9	6

Code

```
bool IsStrobogrammatic(const string& num){
    unordered_map<char,char> dict = {{'0','0'}, {'1','1'}, {'8','8'}, {'6','9'}, {'9','6'}};
    int left = 0, right = num.size() - 1;

    while (left <= right) {
        if (dict.find(num[left] == dict(). end() || dict[num[left]] != num[right]) {
            return false;
        }
        left++;
        right--;
    }
    return true;
}
```

Time Complexity

↳ O(n), where n is the length of the input string.

Space Complexity

↳ O(1), because a fixed map is used