1) Two sum

time = O(N),
space = O(N)

input = [2, 7, 11, 15], target = 9.

output = [0, 1]

return its index:

using hasmap

syntax to create a object of Hashmap

1) HashMap<Integer, Integer> map =
new HashMap<>()

2) use for loop for the itertion from 0 to n = lenght of array.

declare a ~~complement~~ complement ) remaing number to reach target

complement = target - num[i];

if (map.containKey(complement) {
return ~~ma~~ new int[] {map.get(complement), i};

↓
it will return indices

map.put(nums[i], i)

↓

Store number or value and its index.

return new int[0]

Dry run

Input = [3, 3], target = 6.

output = [0, 1]

i = 0; 0 < 2; i

complement = 6 - 3

= 3

check if map.contains 3

if return index of map value, i.e. i

else

map.put(3, 0)

i = 1; 1 < 2; i

complement = 6 - 3

= 3

map already contain 3

return (0, 1)