Two Sum

Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

Example 1:
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2:
Input: nums = [3,2,4], target = 6
Output: [1,2]

Example 3:
Input: nums = [3,3], target = 6
Output: [0,1]

```
twoSum( nums , target){
    Map<Integer, Integer> map = new HashMap<>();
    for( int i=0; i < nums.length; i++){
        int cur = nums[i];
        int x = target - cur;
        if (map.containsKey(x)){
            return new int[]{ map.get(x), i };
        }
        map.put(cur, i);
    }
    return null;
}
```

create HashMap

cur + x = target

if x exist in HashMap then return the index

else add the current index in HashMap

time complexity = O(n)
space complexity = O(n)