**ARRAY**

1. **TWO SUM:**

**Brute Force :**

class Solution {

public int[] twoSum(int[] nums, int target) {

int[] a = new int[2];

for(int i=0; i<nums.length-1; i++)

{

for(int j=0; j<nums.length; j++)

{

if( nums[i] + nums[j] == target && i!=j)

{

a[0] =i;

a[1] =j;

}

}

}

return a;

}

}

**Optimized :**

class Solution {

public int[] twoSum(int[] nums, int target) {

int n = nums.length;

Map<Integer,Integer> map = new HashMap<>();

for(int i=0;i<nums.length;i++){

map.put(nums[i],i);

}

for(int i=0;i<nums.length;i++){

int num = nums[i];

int rem = target - num;

if(map.containsKey(rem)){

int index = map.get(rem);

return new int[] {i,index};

}

}

return new int[] {};

}

}

class Solution {

public int[] twoSum(int[] nums, int target) {

int n = nums.length;

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

for(int i=0;i<nums.length;i++){

if(map.containsKey(target - nums[i]))

return new int[] {i,map.get(target - nums[i])};

map.put(nums[i],i);

}

return new int[] {};

}

}