**SOLUTION**

class Solution {

public:

Solution(){

ios::sync\_with\_stdio(false);

std::cin.tie(nullptr);

std::cout.tie(nullptr);

}

bool containsDuplicate(vector<int>& nums) {

/\* unordered\_map<int,int> m;

for(int i:nums){

if(m[i]>=1){

m[i]++;

return true;

}

else

m[i] = 1;

}

return false;

O(N),O(N)

\*/

if(nums.size() == 0)

return false;

sort(nums.begin(), nums.end());

for(auto i=0; i< nums.size() -1; i++){

if(nums[i] == nums[i+1])

return true;

}

return false;

}

};

**TIME COMPLEXITY: O(N\*logN)**

**SPACE COMPLEXITY: O(1)**

**USING MAP(UNORDERED\_MAP)**

**TC - O(N)**

**SC - O(N)**