Lecture 40 HashMap

1. <https://leetcode.com/problems/contains-duplicate/>

class Solution {

public:

bool containsDuplicate(vector<int>& nums) {

unordered\_map<int, int> m;

for(auto i:nums)

{

m[i]++;

}

for(auto i:m)

{

if(i.second >= 2)

{

return true;

}

}

return false;

}

};

1. <https://leetcode.com/problems/intersection-of-two-arrays-ii/>

class Solution {

public:

vector<int> intersect(vector<int>& nums1, vector<int>& nums2) {

unordered\_map<int, int> freq;

vector<int> ans;

for(auto i:nums1)

{

freq[i]++;

}

for(auto i:nums2)

{

if(freq[i] > 0)

{

ans.push\_back(i);

freq[i]--;

}

}

return ans;

}

};

1. <https://leetcode.com/problems/valid-anagram/>

class Solution {

public:

bool isAnagram(string s, string t) {

map<char, int> mps;

map<char, int> mpt;

for(auto i:s)

{

mps[i]++;

}

for(auto i:t)

{

mpt[i]++;

}

return mps==mpt;

}

};

1. <https://leetcode.com/problems/two-sum/>

class Solution {

public:

vector<int> twoSum(vector<int>& nums, int target) {

unordered\_map<int, int> mp;

int n = nums.size();

for(int i=0; i<n; i++)

{

int diff = target - nums[i];

if(mp.count(diff))

{

return {i, mp[diff]};

}

mp[nums[i]] = i;

}

return {};

}

};