|  |  |
| --- | --- |
| **Experiment-1.4** | |
| **Student Name: Shubham Kumar** | **UID: 21BCS7104** |
| **Branch: CSE** | **Section/Group: 614-A** |
| **Semester: 6th** | **Date of Performance: 07-02-2024** |
| **Subject Name: Advanced Programming Lab-2** | **Subject Code: 21CSP-351** |

**1. Aim:** Hashing.

A) Given a pattern and a string s, find if s follows the same pattern.(290. word pattern)

B) Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.(268. Missing Number)

**2. Source Code/Output:**

**A) Code:**

class Solution {

public:

bool wordPattern(string pattern, string s) {

istringstream is(s);

vector<string> ws;

while (is >> s) {

ws.push\_back(s);

}

if (pattern.size() != ws.size()) {

return false;

}

unordered\_map<char, string> d1;

unordered\_map<string, char> d2;

for (int i = 0; i < ws.size(); ++i) {

char a = pattern[i];

string b = ws[i];

if ((d1.count(a) && d1[a] != b) || (d2.count(b) && d2[b] != a)) {

return false;

}

d1[a] = b;

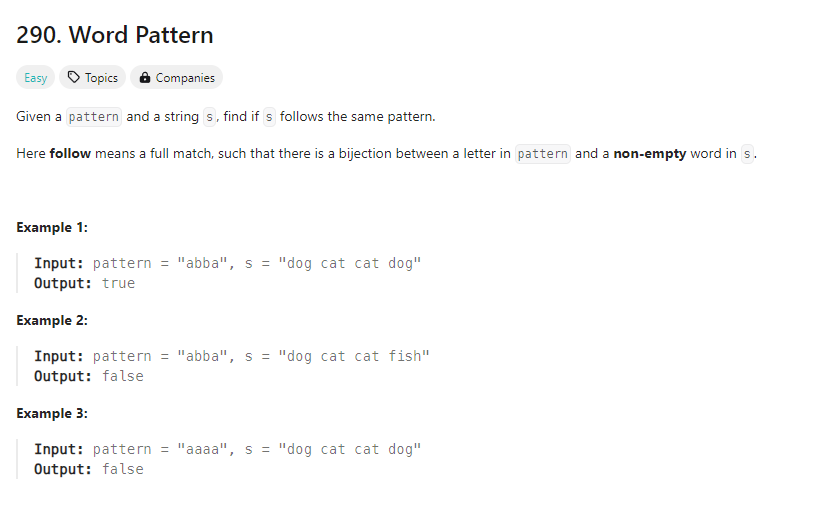
d2[b] = a;

}

return true;

}

};

**Description:**

**Output:**



**B)** **Code:**

class Solution {

public:

int missingNumber(vector<int>& nums) {

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

int res = -1;

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

if(nums[i] != i){

res = i;

break;

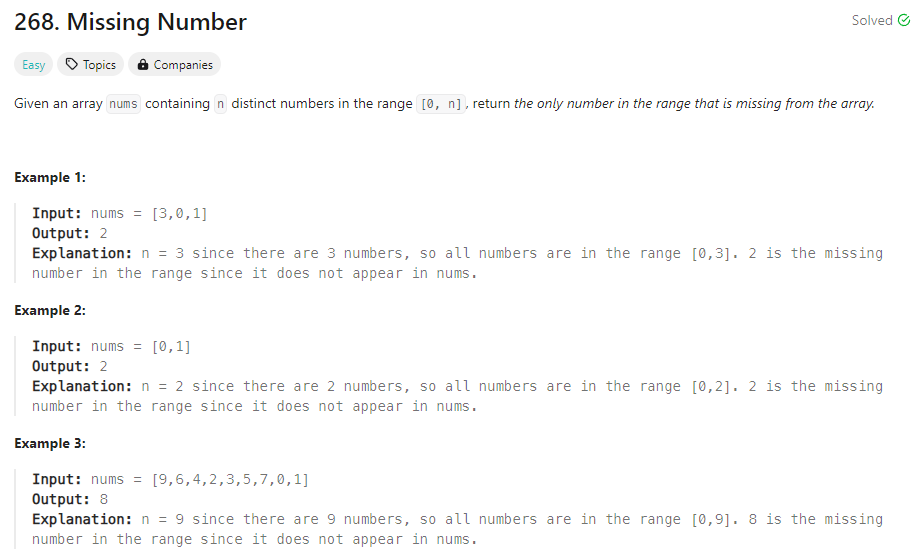
}

return res == -1 ? nums.size() : res;

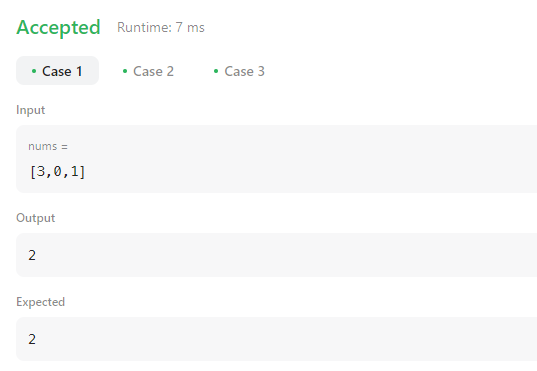
}

};

**Description:**



**Output:**

****