**[Contiguous Array](https://leetcode.com/problems/contiguous-array/)**

Given a binary array nums, return *the maximum length of a contiguous subarray with an equal number of*0*and*1.

**Example 1:**

**Input:** nums = [0,1]

**Output:** 2

**Explanation:** [0, 1] is the longest contiguous subarray with an equal number of 0 and 1.

**Example 2:**

**Input:** nums = [0,1,0]

**Output:** 2

**Explanation:** [0, 1] (or [1, 0]) is a longest contiguous subarray with equal number of 0 and 1.

**Constraints:**

* 1 <= nums.length <= 105
* nums[i] is either 0 or 1.

class Solution {

public:

int findMaxLength(vector<int>& nums) {

int n = nums.size();

unordered\_map<int, int> mp;

int sum = 0;

int subArray\_length = 0;

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

sum += nums[i] == 0 ? -1 : 1;

if (sum == 0) {

subArray\_length = i + 1;

} else if (mp.find(sum) != mp.end()) {

subArray\_length = max(subArray\_length, i - mp[sum]);

} else {

mp[sum] = i;

}

}

return subArray\_length;

}

};

Link : <https://leetcode.com/problems/contiguous-array/?envType=daily-question&envId=2024-03-16>