**Experiment Number: 02**

**Problem Statement: Time complexity analysis of various algorithms to find Majority element**

**Write pseudocode**

**Time complexity analysis derivation**

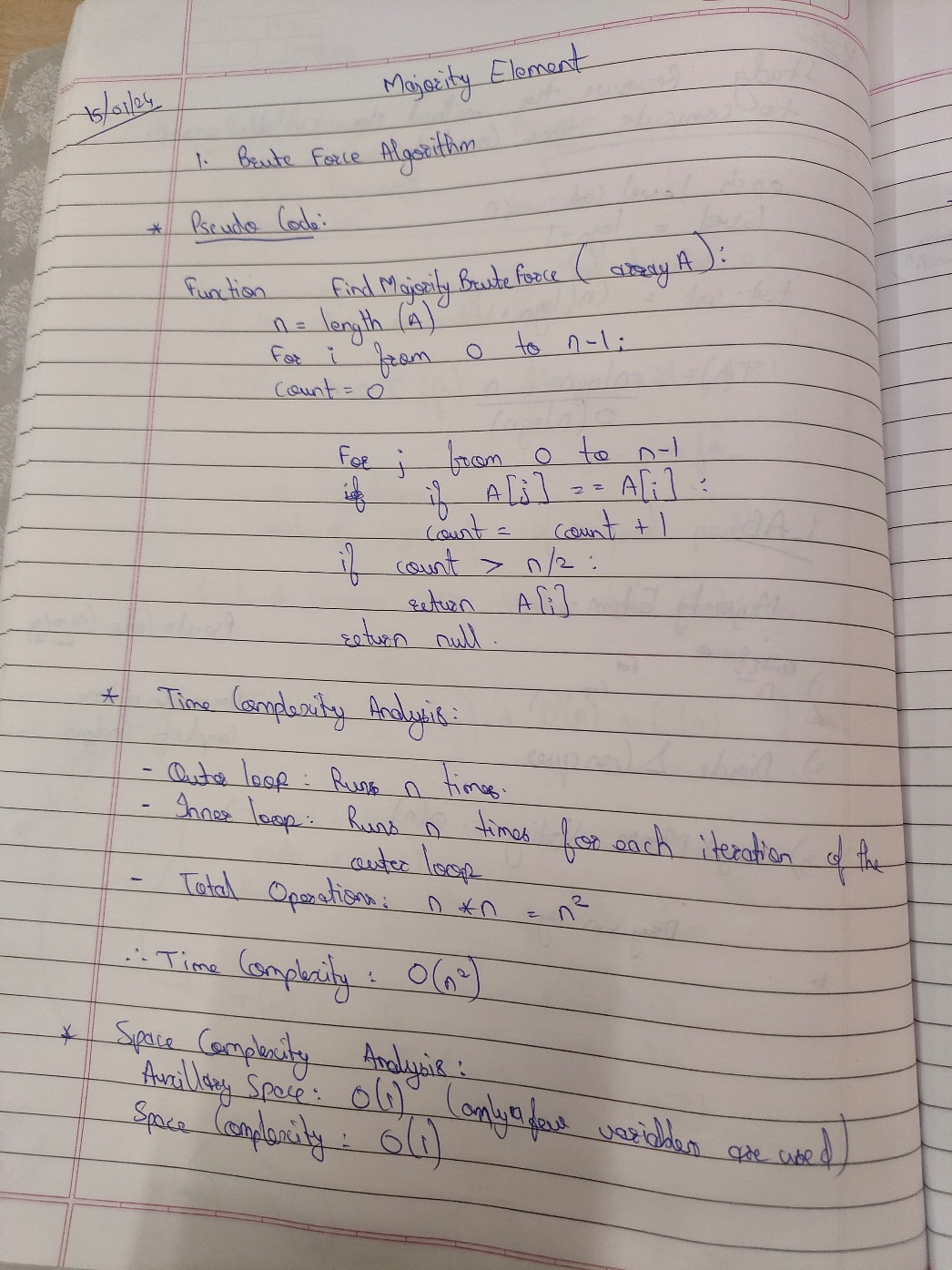
**implemetation of best algorithm**

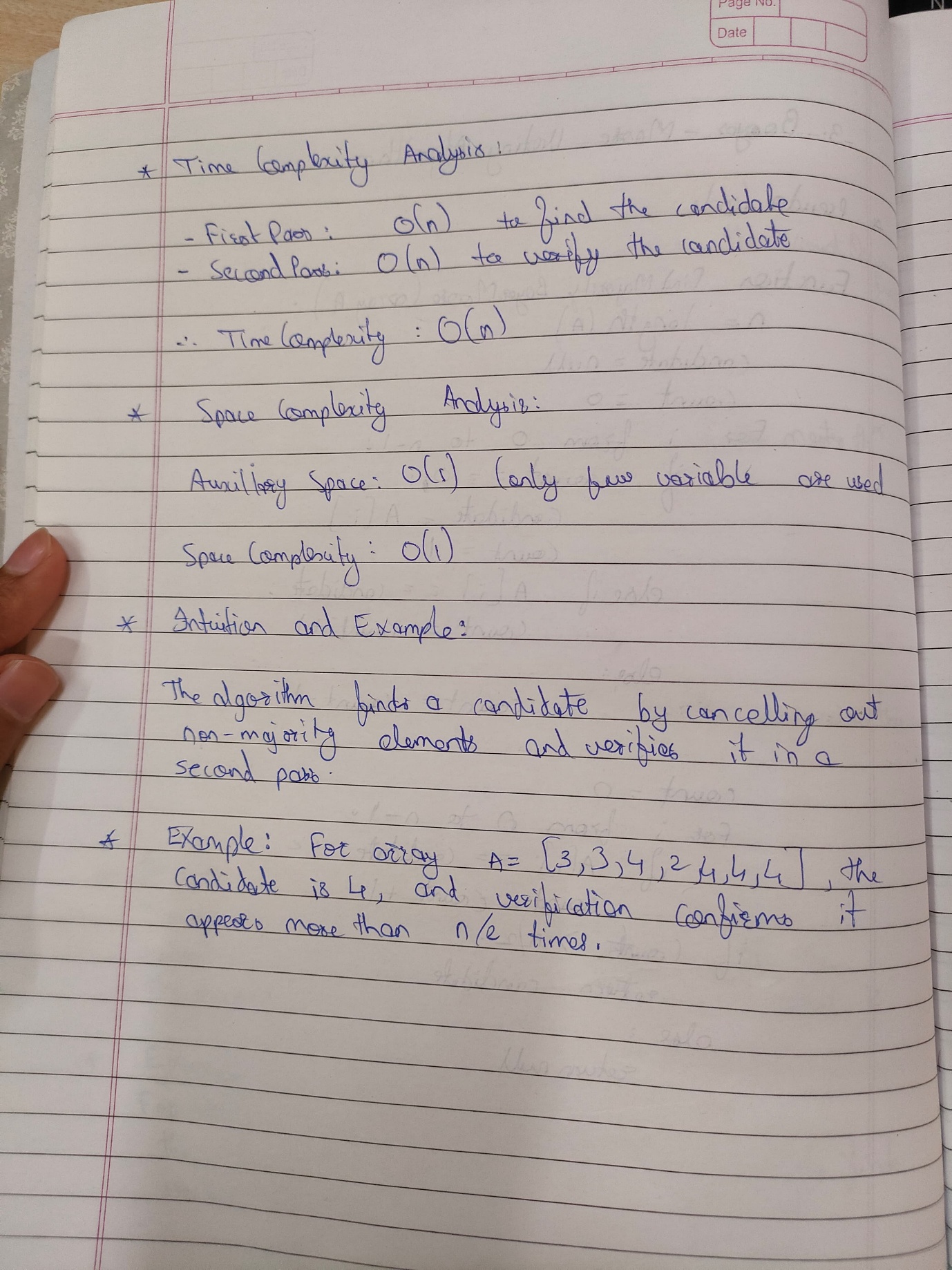
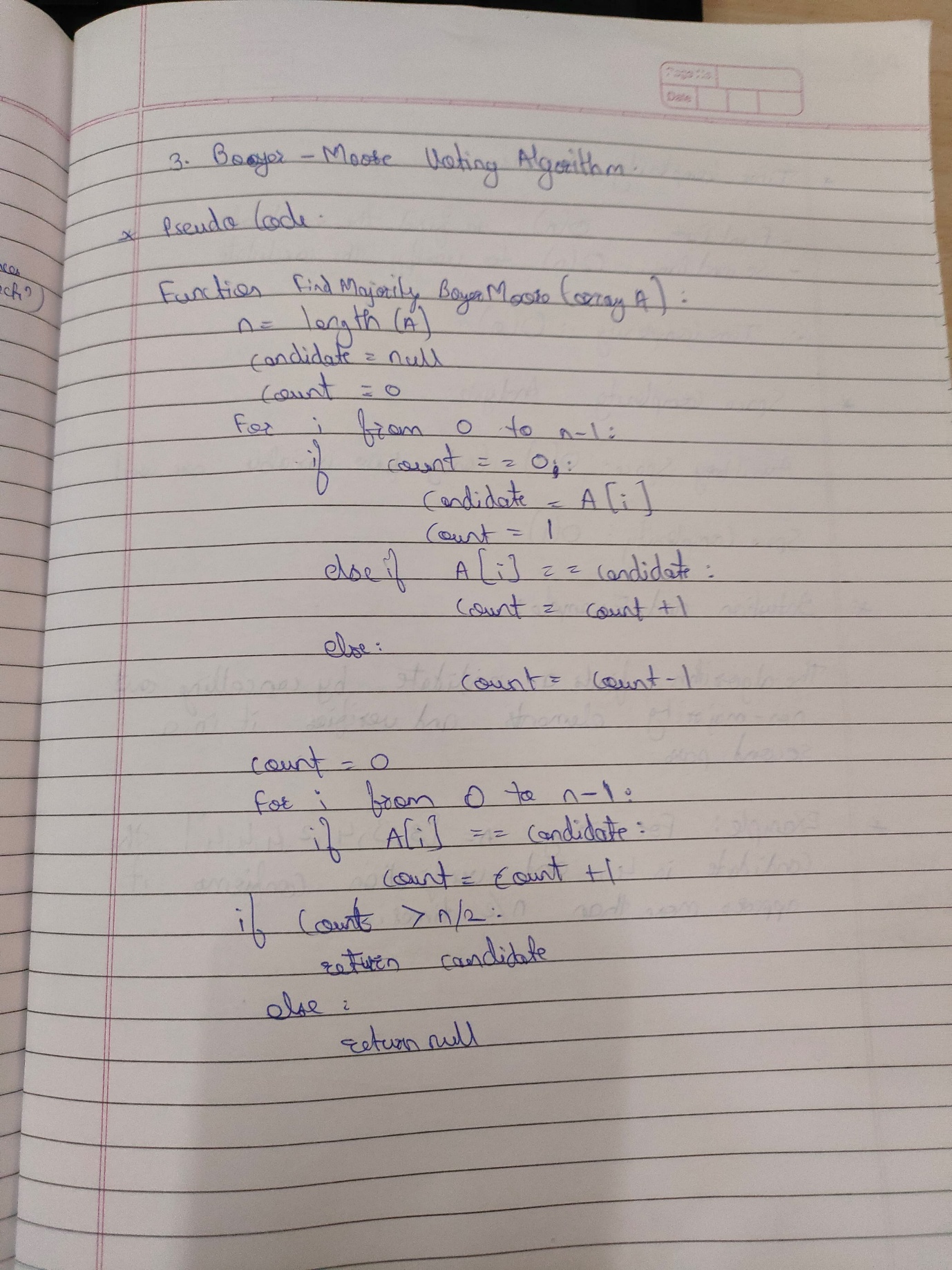
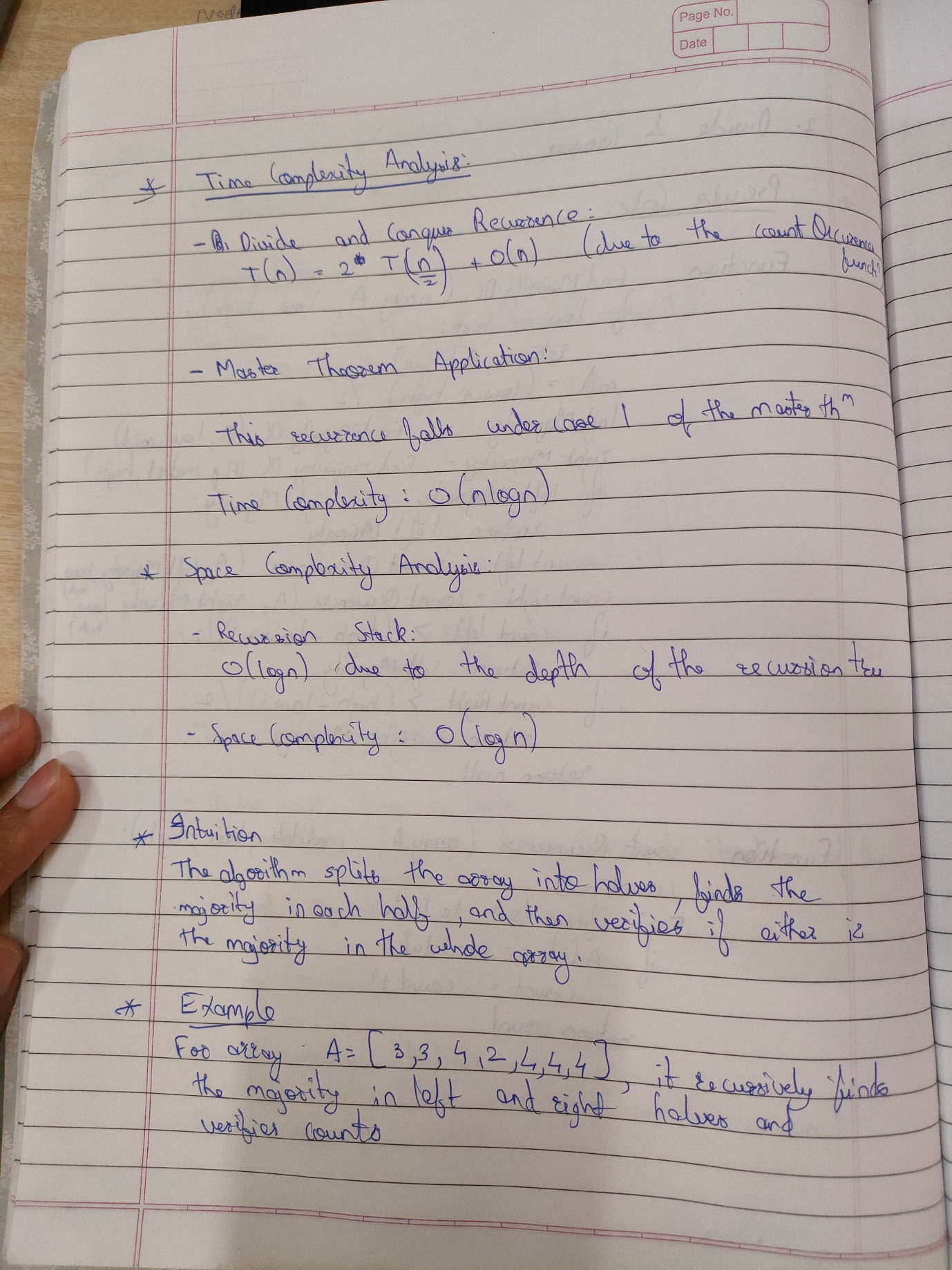
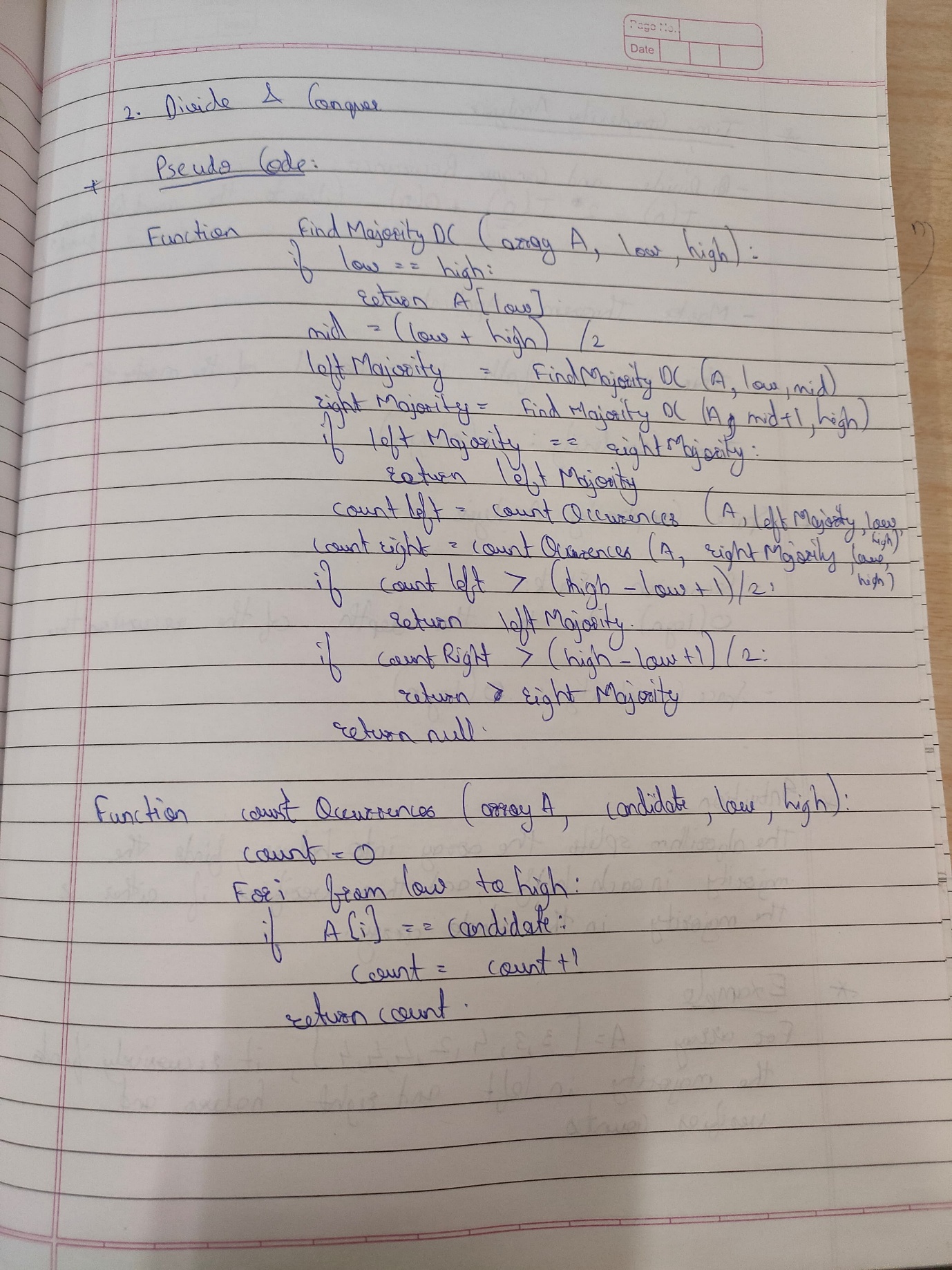
**NAME:** Harshwardhan Patil  **ROLLNO:** 50

**CLASS:** TY-IT-B **BATCH:** B1

**DATE OF PERFORMANCE:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



  
  
Implementation of Boyer-Moore Algorithm:

#include <iostream>

#include <vector>

using namespace std;

int findMajorityElement(const vector<int>& nums) {

int candidate = 0, count = 0;

// Step 1: Find the candidate for the majority element

for (int num : nums) {

if (count == 0) {

candidate = num;

count = 1;

} else if (num == candidate) {

count++;

} else {

count--;

}

}

// Step 2: Verify if the candidate is the majority element

count = 0;

for (int num : nums) {

if (num == candidate) {

count++;

}

}

if (count > nums.size() / 2) {

return candidate;

}

// No majority element found

return -1; // Return -1 to indicate no majority element exists

}

int main() {

vector<int> nums = {2, 2, 1, 1, 1, 2, 2};

int majorityElement = findMajorityElement(nums);

if (majorityElement != -1) {

cout<<"The majority element is: "<<majorityElement<<endl;

} else {

cout << "No majority element exists in the array." << endl;

}

return 0;

}