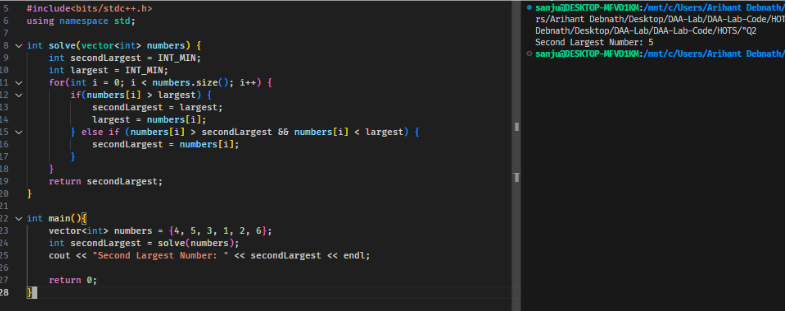
|  | |
| --- | --- |
| Insert Photo | **Name: Arpit Kumar**  **Register Number: RA2211003010142**  **Mail ID: ak0141@srmist.edu.in**  **Department: COMPUTING TECHNOLOGIES**  **Semester: IV** |
| **Subject Title: 21CSC204J Design and Analysis of Algorithm**  **Handled By: N Arunachalam** | |
| **ElAb Completion Status** | |
| **Lab Experiment Completion status** | |
| **REAL WORLD APPLICATION IN DAA PPT VR/SIMULATION DEMO** | |
| **NPTEL/HOTS Questions Solution.** | |

|  | Q1. Given a list of 2n distinct numbers, determine whether the list can be grouped into n  groups of two elements each such that the sum of the pairs in each group is the same.  Code:  #include<bits/stdc++.h>  using namespace std;bool solve(vector<int> numbers) {  int size = numbers.size();  int totalSum = accumulate(numbers.begin(), numbers.end(), 0);  int pairSum = (2\*totalSum)/size; // sum of each pair  // Two Pointer Approach + Sorting  sort(numbers.begin(), numbers.end());  int start = 0;  int end = size - 1;  while(start <= end) {  if(numbers[start] + numbers[end] != pairSum){  return false;  }  start++;  end--;  }  return true;  }  int main(){  vector<int> numbers = {4, 5, 3, 1, 2, 6}; // True  // vector<int> numbers = {1, 1, 1, 1, 1, 1, 2, 3}; // False  if(solve(numbers)) {  cout << "True" << endl;  }  else cout << "False" << endl;  return 0;  }  /\*  Time Complexity : O(NlogN)  Space Complexity : O(1)\*/  Output :  True | | | | | | | | | | | | |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  | | | | | | | | | | | | | | | |

****

**Q2. Give an algorithm to find the second largest element from a list of n numbers.**

**Focus on minimizing the number of comparisons (Hint: there is an algorithm that takes**

**n + o(n) comparisons).**

**Code:**

**#include<bits/stdc++.h>**

**using namespace std;**

**int solve(vector<int> numbers) {**

**int secondLargest = INT\_MIN;**

**int largest = INT\_MIN;**

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

**if(numbers[i] > largest) {**

**secondLargest = largest;**

**largest = numbers[i];**

**} else if (numbers[i] > secondLargest && numbers[i] < largest) {**

**secondLargest = numbers[i];**

**}**

**}**

**return secondLargest;**

**}**

**int main(){**

**vector<int> numbers = {4, 5, 3, 1, 2, 6};**

**int secondLargest = solve(numbers);**

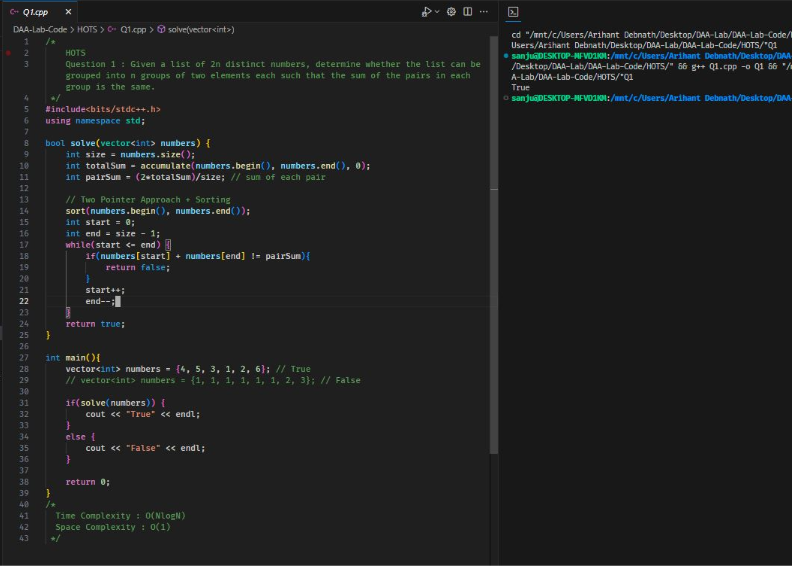
**cout << "Second Largest Number: " << secondLargest << endl;**

**return 0;**

**}**

**Output:**

**Second Largest Number: 5**

****