# END – TERM EXAM

# Algorithm Design & Analysis

Name: - Ankit Kumar

Scholar no.: - 211112004

Sec: - CSE-1

Date: - 18-04-2023

**Ques1.** You are given an unsorted array A [] and a value ‘n’. Your task is to find a pair of elements for which the difference would equal to ‘n’ by using Divide and Conquer Algorithm. Also determine the no. of programs required in order to find such pair and graph of ‘n’ vs varying size of the given array.

```
// Name: - Ankit Kumar
// Scholar no.: - 211112004
// Sec: - CSE-1
// Date: 18-04-2023

#include <bits/stdc++.h>
using namespace std;

// find array of differences of every possible pairs of elements
// in the given array

vector<pair<int, pair<int, int>>> findDiffArray(int A[], int size) {
    vector<pair<int, pair<int, int>>> diffArray;

    // calculate differences
    for(int i = 0; i < size; ++i) {
        for(int j = 0; j < size; ++j) {
            diffArray.push_back({abs(A[i] - A[j]), {A[i], A[j]}});
        }
    }
    return diffArray;
}
```

```cpp
// find pair of elements that will give the difference eqaul to n

pair<int, int> findDiffPairs(int A[], int size, int n) {
    vector<pair<int, pair<int, int>>> diffA = findDiffArray(A, size);

    sort(diffA.begin(), diffA.end()); // sort the diffArray

    // find n as the difference of any pair of elements
    // from the diffArray using Divide and Conquer Algorithm

    int low = 0;
    int high = diffA.size() - 1;

    while(high - low > 1) {
        // find mid of the diffArray
        int mid = (low + high) / 2;

        // decide subarray on which the same algorithm would be implemented
        if(diffA[mid].first <= n) {
            low = mid;
        } else {
            high = mid - 1;
        }
    }

    if(diffA[low].first == n) {
        return diffA[low].second;  // if n found at low position
    } else if(diffA[high].first == n) {
        return diffA[high].second; // if n found at high position
    } else {
        // if no such pair exists then return INT_MAX as pairs
        pair<int, int> p = {INT_MAX, INT_MAX};
        return p;
    }
}

int main() {
    int n;
    cout << "Enter no. of elements in the given array: ";
    cin >> n;
    // take input int the array
    int A[n];
    cout << "Enter " << n << " elements one by one" << endl;
    for(int i = 0; i < n; ++i) cin >> A[i];

    // given diff that would be searched
    int k;
```

```
    cout << "Enter the value of difference that you want pair of elements for:
";
    cin >> k;

    // find the required pair
    pair<int, int> ans = findDiffPairs(A, n, k);

    if(ans.first == INT_MAX && ans.second == INT_MAX) {
        cout << "No such pair exists!" << endl;
    } else {
        cout << "The required pair is : ";
        cout << ans.first << " " << ans.second << endl;
    }
    return 0;
}
```

## Output: -

```
Enter no. of elements in the given array: 10
Enter 10 elements one by one
7 6 0 2 8 12 4 23 9 11
Enter the value of difference that you want pair of elements for: 11
The required pair is : 12 23
```