 Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: DAA (01CT0512)	AIM: Min – Max Problem	
Experiment No: 8	Date: 22/8/2023	Enrolment No: 92100133020

Min – Max Problem:

To find the maximum and minimum numbers in a given array numbers[] of size n, the following algorithm can be used. First, we are representing the naive method and then we will present divide and conquer approach. Naïve method is a basic method to solve any problem. In this method, the maximum and minimum number can be found separately. To find the maximum and minimum numbers, the following straightforward algorithm can be used.

Algorithm:

```

max := numbers[1]
min := numbers[1]

for i = 2 to n do
    if numbers[i] > max then
        max := numbers[i]
    if numbers[i] < min then
        min := numbers[i]
return (max, min)


```

Code:

```

#include <iostream>
using namespace std;
struct Pair {
    int max;
    int min;
};
// Function to find maximum and minimum using the naive algorithm
Pair maxMinNaive(int arr[], int n) {
    Pair result;
    result.max = arr[0];
    result.min = arr[0];
    // Loop through the array to find the maximum and minimum values
    for (int i = 1; i < n; i++) {
        if (arr[i] > result.max) {
            result.max = arr[i]; // Update the maximum value if a larger element is found
        }
        if (arr[i] < result.min) {

```

 Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: DAA (01CT0512)	AIM: Min – Max Problem	
Experiment No: 8	Date: 22/8/2023	Enrolment No: 92100133020

```

        result.min = arr[i]; // Update the minimum value if a smaller element is found
    }
}
return result; // Return the pair of maximum and minimum values
}
int main() {
    int arr[] = {6, 4, 26, 14, 33, 64, 46};
    int n = sizeof(arr) / sizeof(arr[0]);
    Pair result = maxMinNaive(arr, n);
    cout << "Maximum element is: " << result.max << endl;
    cout << "Minimum element is: " << result.min << endl;
    return 0;
}

```

Output:

```

--stdout=Microsoft-MIEngine-
id-nezta2lp.zdu' '--dbgExe=D:
Maximum element is: 64
Minimum element is: 4
PS D:\Mirror\ICT\3rd YEAR\SEM

```

Space complexity: _____

Justification: _____

Time complexity:

Best case time complexity: _____

Justification: _____

Worst case time complexity: _____

Justification: _____