Minimum Difference in an Array

Find minimum difference between any two elements (pair) in given array

Given an unsorted array, find the minimum difference between any pair in the given array.

Examples :

> ***Input:*** *{1, 5, 3, 19, 18, 25}*
> ***Output:*** *1*
> ***Explanation:*** *Minimum difference is between 18 and 19*
>
> ***Input:*** *{30, 5, 20, 9}*
> ***Output:*** *4*
> ***Explanation:*** *Minimum difference is between 5 and 9*
>
> ***Input:*** *{1, 19, -4, 31, 38, 25, 100}*
> ***Output:*** *5*
> ***Explanation:*** *Minimum difference is between 1 and -4*

Naive Approach: To solve the problem follow the below idea:

> *A simple solution is to use two loops two generate every pair of elements and compare them to get*

the minimum difference

Below is the implementation of the above approach:

C++ Java

```
// C++ implementation of simple method to find
// minimum difference between any pair
#include <bits/stdc++.h>
using namespace std;

// Returns minimum difference between any pair
int findMinDiff(int arr[], int n)
{
    // Initialize difference as infinite
    int diff = INT_MAX;

    // Find the min diff by comparing difference
    // of all possible pairs in given array
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (abs(arr[i] - arr[j]) < diff)
                diff = abs(arr[i] - arr[j]);

    // Return min diff
    return diff;
}
```

```
// Driver code
int main()
{
    int arr[] = { 1, 5, 3, 19, 18, 25 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // Function call
    cout << "Minimum difference is " << findMinDiff(arr, n);
    return 0;
}
```

Output

```
Minimum difference is 1
```

Time Complexity: O(N^2).
Auxiliary Space: O(1)

Efficient Approach

C++ Java

```
// C++ program to find minimum difference between
// any pair in an unsorted array
#include <bits/stdc++.h>
```

```
using namespace std;

// Returns minimum difference between any pair
int findMinDiff(int arr[], int n)
{
    // Sort array in non-decreasing order
    sort(arr, arr + n);

    // Initialize difference as infinite
    int diff = INT_MAX;

    // Find the min diff by comparing adjacent
    // pairs in sorted array
    for (int i = 0; i < n - 1; i++)
        if (arr[i + 1] - arr[i] < diff)
            diff = arr[i + 1] - arr[i];

    // Return min diff
    return diff;
}

// Driver code
int main()
{
    int arr[] = { 1, 5, 3, 19, 18, 25 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // Function call
    cout << "Minimum difference is " << findMinDiff(arr, n);
```

```
    return 0;
}
```


Output

```
Minimum difference is 1
```

Time Complexity: O(N log N)
Auxiliary Space: O(1)