Find minimum difference between any two elements

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

Examples :

```
Input  : {1, 5, 3, 19, 18, 25};
Output : 1
Minimum difference is between 18 and 19

Input  : {30, 5, 20, 9};
Output : 4
Minimum difference is between 5 and 9

Input  : {1, 19, -4, 31, 38, 25, 100};
Output : 5
Minimum difference is between 1 and -4
```

We strongly recommend that you click here and practice it, before moving on to the solution.

Method 1 (Simple: O(n^2)

A simple solution is to use two loops.

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

Output :

```
Minimum difference is 1
```

Method 2 (Efficient: O(n Log n)

The idea is to use sorting. Below are steps.

1) Sort array in ascending order. This step takes O(n Log n) time.
2) Initialize difference as infinite. This step takes O(1) time.
3) Compare all adjacent pairs in sorted array and keep track of minimum difference. This step takes O(n) time.

Below is C++ implementation of above idea.

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

Output :

```
Minimum difference is 1
```