



[Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Topic-wise Practice](#) [C++](#) [Java](#) [Python](#) [Competitive Programming](#) [Machi](#)

# Find minimum difference between any two elements

Difficulty Level : Basic • Last Updated : 21 Jan, 2022

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



## Start Your Coding Journey Now!

[Login](#)[Register](#)

### Method 1 (Simple: $O(n^2)$ )

A simple solution is to use two loops.

### C++

```
// 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;
}
```



# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
return 0;  
}
```

## Java

```
// Java implementation of simple method to find  
// minimum difference between any pair  
  
class GFG  
{  
    // Returns minimum difference between any pair  
    static int findMinDiff(int[] arr, int n)  
    {  
        // Initialize difference as infinite  
        int diff = Integer.MAX_VALUE;  
  
        // 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 (Math.abs((arr[i] - arr[j])) < diff)  
                    diff = Math.abs((arr[i] - arr[j]));  
  
        // Return min diff  
        return diff;  
    }  
}
```



# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
}  
}
```

## Python3

```
# Python implementation of simple method to find  
# minimum difference between any pair
```

```
# Returns minimum difference between any pair
```

```
def findMinDiff(arr, n):  
    # Initialize difference as infinite  
    diff = 10**20  
  
    # Find the min diff by comparing difference  
    # of all possible pairs in given array  
    for i in range(n-1):  
        for j in range(i+1,n):  
            if abs(arr[i]-arr[j]) < diff:  
                diff = abs(arr[i] - arr[j])
```

```
# Return min diff  
return diff
```

```
# Driver code  
arr = [1, 5, 3, 19, 18, 25]  
n = len(arr)
```

## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// C# implementation of simple method to find
// minimum difference between any pair
using System;

class GFG {

    // Returns minimum difference between any pair
    static int findMinDiff(int []arr, int n)
    {

        // Initialize difference as infinite
        int diff = int.MaxValue;

        // 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 (Math.Abs((arr[i] - arr[j])) < diff)
                    diff = Math.Abs((arr[i] - arr[j]));

        // Return min diff
        return diff;
    }

    // Driver method to test the above function
    public static void Main()
    {
```

# Start Your Coding Journey Now!

[Login](#)[Register](#)

// This code is contributed by nitin mittal.

## PHP

```
<?php
// PHP implementation of simple
// method to find minimum
// difference between any pair

// Returns minimum difference
// between any pair
function findMinDiff($arr, $n)
{
    // Initialize difference
    // as infinite
    $diff = PHP_INT_MAX;

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

    // Return min diff
    return $diff;
```

# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
findMinDiff($arr, $n);
```

```
// This code is contributed by ajit  
?>
```

## Javascript

```
<script>
```

```
// JavaScript program implementation of simple method to find  
// minimum difference between any pair
```

```
// Returns minimum difference between any pair  
function findMinDiff( arr, n)  
{  
    // Initialize difference as infinite  
    let diff = Number.MAX_VALUE;  
  
    // Find the min diff by comparing difference  
    // of all possible pairs in given array  
    for (let i=0; i<n-1; i++)  
        for (let j=i+1; j<n; j++)  
            if (Math.abs((arr[i] - arr[j])) < diff)  
                diff = Math.abs((arr[i] - arr[j]));  
  
    // Return min diff  
    return diff;  
}
```



## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
findMinDiff(arr, arr.length));
```

```
</script>
```

### 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 implementation of above idea.

### C++



```
// C++ program to find minimum difference between  
// any pair in an unsorted array
```



## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// 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;
}
```



# Java

## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
{  
    // Returns minimum difference between any pair  
    static int findMinDiff(int[] arr, int n)  
    {  
        // Sort array in non-decreasing order  
        Arrays.sort(arr);  
  
        // Initialize difference as infinite  
        int diff = Integer.MAX_VALUE;  
  
        // 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 method to test the above function  
    public static void main(String[] args)  
    {  
        int arr[] = new int[]{1, 5, 3, 19, 18, 25};  
        System.out.println("Minimum difference is "+  
                           findMinDiff(arr, arr.length));  
    }  
}
```



## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
# any pair in an unsorted array

# Returns minimum difference between any pair
def findMinDiff(arr, n):

    # Sort array in non-decreasing order
    arr = sorted(arr)

    # Initialize difference as infinite
    diff = 10**20

    # Find the min diff by comparing adjacent
    # pairs in sorted array
    for i in range(n-1):
        if arr[i+1] - arr[i] < diff:
            diff = arr[i+1] - arr[i]

    # Return min diff
    return diff

# Driver code
arr = [1, 5, 3, 19, 18, 25]
n = len(arr)
print("Minimum difference is " + str(findMinDiff(arr, n)))

# This code is contributed by Pratik Chhajer
```

**C#**

# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
class GFG
{
    // Returns minimum difference
    // between any pair
    static int findMinDiff(int[] arr,
                           int n)
    {
        // Sort array in
        // non-decreasing order
        Array.Sort(arr);

        // Initialize difference
        // as infinite
        int diff = int.MaxValue;

        // 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
    public static void Main()
    {
```



# Start Your Coding Journey Now!

[Login](#)[Register](#)

//This code is contributed by anuj\_67.

## PHP

```
<?php
// PHP program to find minimum
// difference between any pair
// in an unsorted array

// Returns minimum difference
// between any pair
function findMinDiff($arr, $n)
{

// Sort array in
// non-decreasing order
sort($arr);

// Initialize difference
// as infinite
$diff = PHP_INT_MAX;

// Find the min diff by
// comparing adjacent
// pairs in sorted array
for ($i = 0; $i < $n - 1; $i++)
```



# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Driver code
$arr = array(1, 5, 3, 19, 18, 25);
$n = sizeof($arr);
echo "Minimum difference is " ,
    findMinDiff($arr, $n);

// This code is contributed ajit
?>
```

## Javascript

<script>

```
// Javascript program to find minimum
// difference between any pair
// in an unsorted array

// Returns minimum difference
// between any pair
function findMinDiff(arr, n)
{
    // Sort array in
    // non-decreasing order
    arr.sort(function(a, b)
    {return a - b});
```



## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
// Computing minimum pair  
// in sorted array  
for (let i = 0; i < n - 1; i++)  
    if (arr[i + 1] - arr[i] < diff)  
        diff = arr[i + 1] - arr[i];  
  
// Return min diff  
return diff;  
}  
  
let arr = [1, 5, 3, 19, 18, 25];  
document.write("Minimum difference is "  
+ findMinDiff(arr, arr.length));
```

</script>

### Output :

Minimum difference is 1



## Start Your Coding Journey Now!

[Login](#)[Register](#)

Asked in: [Amazon](#)

This article is contributed by **Harshit Agrawal**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.





**Start Your Coding Journey Now!**

Login

Register

Previous

**Size of The Subarray With Maximum Sum**

Next

**Space optimization using bit manipulations**

## RECOMMENDED ARTICLES

Page :

**Article Contributed By :**



GeeksforGeeks



**Vote for difficulty**

## Start Your Coding Journey Now!

[Login](#)[Register](#)

Improved By : [nitin mittal](#), [vt\\_m](#), [jit\\_t](#), [chinmoy1997pal](#), [divyesh072019](#), [amartyani20](#)

Article Tags : [Amazon](#), [Arrays](#), [Sorting](#)

Practice Tags : [Amazon](#), [Arrays](#), [Sorting](#)

[Report Issue](#)

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

[Load Comments](#)

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)



## Start Your Coding Journey Now!

[Login](#)[Register](#)[Careers](#)[In Media](#)[Contact Us](#)[Privacy Policy](#)  
[Copyright Policy](#)

### News

[Top News](#)[Technology](#)[Work & Career](#)[Business](#)[Finance](#)[Lifestyle](#)[Data Structures](#)[SDE Cheat Sheet](#)[Machine learning](#)[CS Subjects](#)  
[Video Tutorials](#)

### Languages

[Python](#)[Java](#)[CPP](#)[Golang](#)[C#](#)[SQL](#)

### Web Development

[Web Tutorials](#)[Django Tutorial](#)[HTML](#)[CSS](#)[JavaScript](#)

### Contribute

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Write Interview Experience](#)[Internships](#)

**Start Your Coding Journey Now!**

Login

Register

