

VIJAY M

22IT125

DAY-7

CODING PRACTICES AND PROBLEM

1.3Sum closest

Time complexity : $O(n^2)$

Solution:

```
import java.util.*;
```

```
class GfG {
```

```
    static List<Integer> closestTripletSum(int[] arr, int target) {  
        int n = arr.length;
```

```
        List<Integer> res = new ArrayList<>(Arrays.asList(0, 0, 0));  
        int minDiff = Integer.MAX_VALUE;
```

```
        for (int i = 0; i < n - 2; i++) {  
            for (int j = i + 1; j < n - 1; j++) {  
                for (int k = j + 1; k < n; k++) {  
                    int currSum = arr[i] + arr[j] + arr[k];  
                    int currDiff = Math.abs(currSum - target);
```

```
                    if (currDiff < minDiff) {  
                        minDiff = currDiff;  
                        res.set(0, arr[i]);  
                        res.set(1, arr[j]);  
                        res.set(2, arr[k]);  
                    }  
                }  
            }  
        }
```

```

    return res;
}

public static void main(String[] args) {
    int[] arr = {1, 2, 3, 4, -5};
    int target = 10;

    List<Integer> res = closestTripletSum(arr, target);
    System.out.println(res.get(0) + " " + res.get(1) + " " + res.get(2));
}
}

```

Output:

The screenshot shows an IDE with the following components:

- Project View:** Shows the project structure with folders for .idea, out, and src. The src folder contains the file TicTacToe.java.
- Code Editor:** Displays the code for TicTacToe.java. The code defines a method `closestTripletSum` that finds the closest triplet sum to a target. The main method calls this function with the array `{1, 2, 3, 4, -5}` and target `10`.
- Run Console:** Shows the command used to run the program: `C:\Users\sanja\jdk-23.0.1\bin\java.exe --javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=52027:C:\Users\sanja\AppData\Local\...`. The output is `2 3 4`.
- Status Bar:** Shows the file path `TRIPLET > src > GfG > closestTripletSum` and the encoding `UTF-8`.

2.Jump Game II

Time Complexity : $O(n^2)$

Solution:

```
import java.io.*;
import java.util.*;

class GFG {

    static int minJumps(int arr[], int l, int h)
    {

        if (h == l)
            return 0;

        if (arr[l] == 0)
            return Integer.MAX_VALUE;

        int min = Integer.MAX_VALUE;
        for (int i = l + 1; i <= h && i <= l + arr[l];
            i++) {
            int jumps = minJumps(arr, i, h);
            if (jumps != Integer.MAX_VALUE
                && jumps + 1 < min)
                min = jumps + 1;
        }
        return min;
    }

    public static void main(String args[])
    {
        int arr[] = { 1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9 };
        int n = arr.length;
```

```

System.out.print(
    "Minimum number of jumps to reach end is "
    + minJumps(arr, 0, n - 1));
}
}

```

Output:

The screenshot shows an IDE with a project named 'TRIPLET'. The file 'TicTacToe.java' is open, containing the following code:

```

1
2 import java.io.*;
3 import java.util.*;
4
5 class GFG {
6
7     static int minJumps(int arr[], int l, int h) {
8         {
9
10            if (h == l)
11                return 0;
12
13
14            if (arr[l] == 0)
15                return Integer.MAX_VALUE;
16
17            int min = Integer.MAX_VALUE;
18            for (int i = l + 1; i <= h && i <= l + arr[l];
19                 i++) {
20                int jumps = minJumps(arr, i, h);

```

The Run window shows the following output:

```

C:\Users\sanja\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=53058:C:\Users\sanja\AppData\Local\J
Minimum number of jumps to reach end is 3
Process finished with exit code 0

```

3.Group Anagrams

Time Complexity : $O(NM \log M + MN \log N)$

Solution:

```

import java.util.*;

public class Main {

    static class Pair implements Comparable<Pair> {
        String x;
        int y;
        public Pair(String x, int y)

```

```

    {
        this.x = x;
        this.y = y;
    }
    public int compareTo(Pair o)
    {
        return this.x.compareTo(o.x);
    }
}

static ArrayList<Pair>
createDuplicateArray(String[] wordArr, int size)
{
    ArrayList<Pair> dupArray = new ArrayList<Pair>();
    for (int i = 0; i < size; i++) {
        Pair p = new Pair(wordArr[i], i);
        dupArray.add(p);
    }
    return dupArray;
}

static void printAnagramsTogether(String[] wordArr,
                                int size)
{
    ArrayList<Pair> dupArray = new ArrayList<Pair>();

    dupArray = createDuplicateArray(
        wordArr, size);

    for (int i = 0; i < size; ++i) {
        Pair e = dupArray.get(i);
        char[] arr = e.x.toCharArray();
        Arrays.sort(arr);
        String x = String.valueOf(arr);
        Pair p = new Pair(x, e.y);
    }
}

```

```

        dupArray.set(i, p);
    }

    Collections.sort(dupArray);

    for (int i = 0; i < size; ++i)
        System.out.print(wordArr[dupArray.get(i).y]
            + " ");
    }

    public static void main(String[] args)
    {
        String[] wordArr
            = { "cat", "dog", "tac", "god", "act" };
        printAnagramsTogether(wordArr, wordArr.length);
    }
}

```

Output:

The screenshot shows an IDE window with a project named 'TRIPILET'. The 'src' folder contains a file named 'Main'. The code in 'Main.java' defines a 'Pair' class and a 'createDuplicateArray' method. The output console shows the execution of the program, which prints the words 'cat tac act dog god' in order.

```

1
2 import java.util.*;
3 public class Main {
4
5     static class Pair implements Comparable<Pair> { 12 usages
6         String x; 4 usages
7         int y; 3 usages
8         public Pair(String x, int y) 2 usages
9         {
10             this.x = x;
11             this.y = y;
12         }
13         public int compareTo(Pair o)
14         {
15             return this.x.compareTo(o.x);
16         }
17     }
18     static ArrayList<Pair> 1 usage
19     createDuplicateArray(String[] wordArr, int size)
20     {

```

```

C:\Users\sanja\jdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=54538:C:\Users\sanja\AppData\Local\J
cat tac act dog god
Process finished with exit code 0

```

4. Decoding Ways

Time Complexity : $O(n)$

Solution:

```
class GFG {  
  
    static int countDecoding(char[] digits, int n)  
    {  
  
        if (n == 0 || n == 1)  
            return 1;  
  
        if (digits[0] == '0')  
            return 0;  
  
        int count = 0;  
  
        if (digits[n - 1] > '0')  
            count = countDecoding(digits, n - 1);  
  
        if (digits[n - 2] == '1'  
            || (digits[n - 2] == '2'  
                && digits[n - 1] < '7'))  
            count += countDecoding(digits, n - 2);  
  
        return count;  
    }  
  
    static int countWays(char[] digits, int n)  
    {  
        if (n == 0 || (n == 1 && digits[0] == '0'))  
            return 0;  
        return countDecoding(digits, n);  
    }  
}
```

```

public static void main(String[] args)
{
    char digits[] = { '1', '2', '3', '4' };
    int n = digits.length;
    System.out.printf("Count is %d",
        countWays(digits, n));
}
}

```

Output:

```

1
2 class GFG {
3
4     static int countDecoding(char[] digits, int n) 3 usages
5     {
6
7         if (n == 0 || n == 1)
8             return 1;
9
10        if (digits[0] == '0')
11            return 0;
12
13        int count = 0;
14
15        if (digits[n - 1] > '0')
16            count = countDecoding(digits, n - 1);
17
18        if (digits[n - 2] == '1'
19            || (digits[n - 2] == '2'
20                && digits[n - 1] < '7'))

```

Run GFG

```

C:\Users\sanja\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=55550:C:\Users\sanja\AppData\Local\J
Count is 3
Process finished with exit code 0

```

5. Best Time to Buy and Sell

Time Complexity : $O(n)$

Solution:

```

import java.util.ArrayList;
import java.util.List;
class GfG {

```



```

static int maxProfit(int[] prices) {
    int minSoFar = prices[0];
    int res = 0;

    for (int i = 1; i < prices.length; i++) {
        minSoFar = Math.min(minSoFar, prices[i]);

        res = Math.max(res, prices[i] - minSoFar);
    }
    return res;
}

public static void main(String[] args) {
    int[] prices = {7, 10, 1, 3, 6, 9, 2};
    System.out.println(maxProfit(prices));
}
}

```

Output:

The screenshot shows an IDE window with a project named 'TRIPILET'. The file 'Main.java' is open, displaying the following code:

```

1  import java.util.ArrayList;
2  import java.util.List;
3
4
5  class GfG {
6
7      @
8      static int maxProfit(int[] prices) { 1 usage
9          int minSoFar = prices[0];
10         int res = 0;
11
12         for (int i = 1; i < prices.length; i++) {
13             minSoFar = Math.min(minSoFar, prices[i]);
14
15             res = Math.max(res, prices[i] - minSoFar);
16         }
17         return res;
18     }
19
20     public static void main(String[] args) {
21         int[] prices = {7, 10, 1, 3, 6, 9, 2};

```

The 'Run' tab shows the command executed: `C:\Users\sanja\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=55893:C:\Users\sanja\AppData\Local\Je`. The output is `8`. The status bar at the bottom indicates the file is 'GfG' and the encoding is 'UTF-8'.

6.Number of Islands

Time Complexity : $O(mn)$

Solution:

```
public class GfG {

    static boolean isSafe(char[][] M, int r, int c,
                          boolean[][] visited) {
        int ROW = M.length;
        int COL = M[0].length;

        return r >= 0 && r < ROW && c >= 0 && c < COL
            && M[r][c] == '1' && !visited[r][c];
    }

    static void DFS(char[][] M, int r, int c,
                    boolean[][] visited) {

        int[] rNbr = { -1, -1, -1, 0, 0, 1, 1, 1 };
        int[] cNbr = { -1, 0, 1, -1, 1, -1, 0, 1 };

        visited[r][c] = true;

        for (int k = 0; k < 8; ++k) {
            int newR = r + rNbr[k];
            int newC = c + cNbr[k];
            if (isSafe(M, newR, newC, visited)) {
                DFS(M, newR, newC, visited);
            }
        }
    }

    static int countIslands(char[][] M) {
        int ROW = M.length;
```

```
int COL = M[0].length;
```

```
boolean[][] visited = new boolean[ROW][COL];
```

```
int count = 0;
```

```
for (int r = 0; r < ROW; ++r) {
```

```
    for (int c = 0; c < COL; ++c) {
```

```
        if (M[r][c] == '1' && !visited[r][c]) {
```

```
            DFS(M, r, c, visited);
```

```
            ++count;
```

```
        }
```

```
    }
```

```
}
```

```
return count;
```

```
}
```

```
public static void main(String[] args) {
```

```
    char[][] M = {
```

```
        { '1', '1', '0', '0', '0' },
```

```
        { '0', '1', '0', '0', '1' },
```

```
        { '1', '0', '0', '1', '1' },
```

```
        { '0', '0', '0', '0', '0' },
```

```
        { '1', '0', '1', '1', '0' }
```

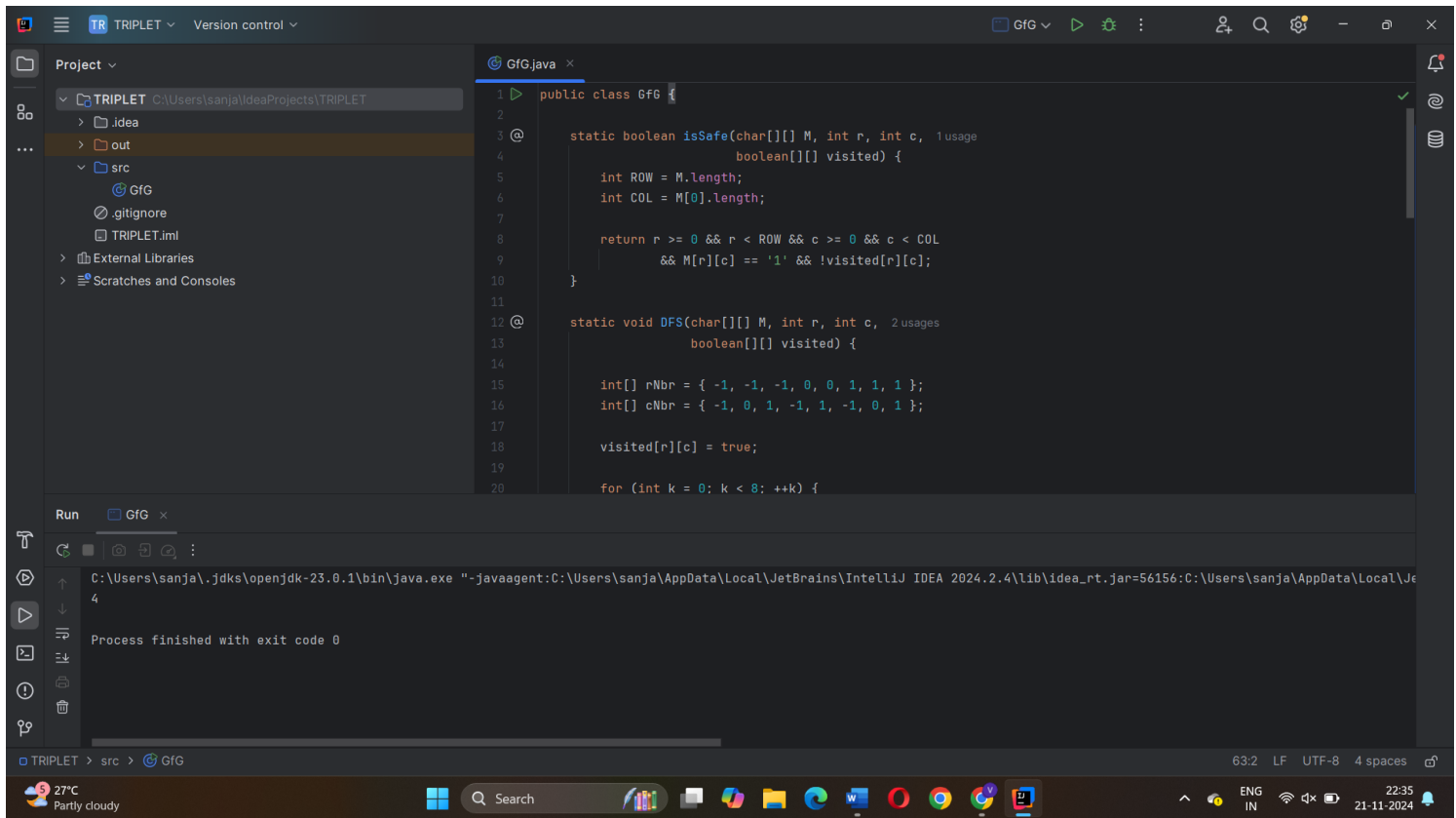
```
    };
```

```
    System.out.println(countIslands(M));
```

```
}
```

```
}
```

Output:



```
1 public class GfG {
2
3     @
4     static boolean isSafe(char[][] M, int r, int c, 1 usage
5         boolean[][] visited) {
6
7         int ROW = M.length;
8         int COL = M[0].length;
9
10        return r >= 0 && r < ROW && c >= 0 && c < COL
11            && M[r][c] == '1' && !visited[r][c];
12    }
13
14    @
15    static void DFS(char[][] M, int r, int c, 2 usages
16        boolean[][] visited) {
17
18        int[] rNbr = { -1, -1, -1, 0, 0, 1, 1, 1 };
19        int[] cNbr = { -1, 0, 1, -1, 1, -1, 0, 1 };
20
21        visited[r][c] = true;
22
23        for (int k = 0; k < 8; ++k) {
```

Run GfG x

C:\Users\sanja\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=56156:C:\Users\sanja\AppData\Local\J...
4

Process finished with exit code 0

7.Quick Sort

Time cpmplexity:

Best Case : ($\Omega(n \log n)$)

Worst Case : ($\theta(n \log n)$)

Average Case : ($O(n^2)$)

Solution:

```
import java.util.Arrays;
```

```
class GfG {
```

```
    static int partition(int[] arr, int low, int high) {
```

```
        int pivot = arr[high];
```

```
int i = low - 1;
```

```
for (int j = low; j <= high - 1; j++) {  
    if (arr[j] < pivot) {  
        i++;  
        swap(arr, i, j);  
    }  
}
```

```
swap(arr, i + 1, high);  
return i + 1;  
}
```

```
static void swap(int[] arr, int i, int j) {  
    int temp = arr[i];  
    arr[i] = arr[j];  
    arr[j] = temp;  
}
```

```
static void quickSort(int[] arr, int low, int high) {  
    if (low < high) {  
  
        int pi = partition(arr, low, high);  
  
        quickSort(arr, low, pi - 1);  
        quickSort(arr, pi + 1, high);  
    }  
}
```

```
public static void main(String[] args) {  
    int[] arr = {10, 7, 8, 9, 1, 5};  
    int n = arr.length;  
  
    quickSort(arr, 0, n - 1);  
}
```

```

    for (int val : arr) {
        System.out.print(val + " ");
    }
}
}
}

```

Output:

```

1  import java.util.Arrays;
2
3  class GfG {
4
5      static int partition(int[] arr, int low, int high) {
6
7          int pivot = arr[high];
8
9          int i = low - 1;
10
11         for (int j = low; j <= high - 1; j++) {
12             if (arr[j] < pivot) {
13                 i++;
14                 swap(arr, i, j);
15             }
16         }
17
18         swap(arr, i + 1, high);
19         return i + 1;
20     }
21 }

```

Run GfG

```

C:\Users\sanja\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=56447:C:\Users\sanja\AppData\Local\J
1 5 7 8 9 10
Process finished with exit code 0

```

8.Merge Sort

Time Complexity:

Best Case : $O(n \log n)$

Worst Case : $O(n \log n)$

Average Case : $O(n \log n)$

Solution:

```
import java.io.*;
```

```
class GfG {
```

```
    static void merge(int arr[], int l, int m, int r)
    {
```

```
        int n1 = m - l + 1;
```

```
        int n2 = r - m;
```

```
        int L[] = new int[n1];
```

```
        int R[] = new int[n2];
```

```
        for (int i = 0; i < n1; ++i)
```

```
            L[i] = arr[l + i];
```

```
        for (int j = 0; j < n2; ++j)
```

```
            R[j] = arr[m + 1 + j];
```

```
        int i = 0, j = 0;
```

```
        int k = l;
```

```
        while (i < n1 && j < n2) {
```

```
            if (L[i] <= R[j]) {
```

```
                arr[k] = L[i];
```

```
                i++;
```

```
            }
```

```
            else {
```

```
                arr[k] = R[j];
```

```
                j++;
```

```
            }
```

```
            k++;
```

```
        }
```

```
while (i < n1) {  
    arr[k] = L[i];  
    i++;  
    k++;  
}
```

```
while (j < n2) {  
    arr[k] = R[j];  
    j++;  
    k++;  
}  
}
```

```
static void sort(int arr[], int l, int r)  
{  
    if (l < r) {  
  
        int m = l + (r - l) / 2;  
  
        sort(arr, l, m);  
        sort(arr, m + 1, r);  
  
        merge(arr, l, m, r);  
    }  
}
```

```
static void printArray(int arr[])  
{  
    int n = arr.length;  
    for (int i = 0; i < n; ++i)  
        System.out.print(arr[i] + " ");  
    System.out.println();  
}
```

```
public static void main(String args[])
```



```

    }
    int arr[] = { 12, 11, 13, 5, 6, 7 };

    System.out.println("Given array is");
    printArray(arr);

    sort(arr, 0, arr.length - 1);

    System.out.println("\nSorted array is");
    printArray(arr);
}
}

```

Output:

The screenshot shows an IDE window with a project named 'TRIPILET'. The code in 'GfG.java' implements a merge sort algorithm. The output window shows the execution results:

```

Run GfG x
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13

Process finished with exit code 0

```

The code in the editor is as follows:

```

1
2 import java.io.*;
3
4 class GfG {
5
6     static void merge(int arr[], int l, int m, int r) {
7         {
8
9             int n1 = m - l + 1;
10            int n2 = r - m;
11
12            int L[] = new int[n1];
13            int R[] = new int[n2];
14
15            for (int i = 0; i < n1; ++i)
16                L[i] = arr[l + i];
17            for (int j = 0; j < n2; ++j)
18                R[j] = arr[m + 1 + j];
19
20

```

9.Ternary Search

Time Complexity : $O(2 * \log_3 n)$

Solution:

```
class GFG {
    static int ternarySearch(int l, int r, int key, int ar[])
    {
        while (r >= l) {
            int mid1 = l + (r - l) / 3;
            int mid2 = r - (r - l) / 3;

            if (ar[mid1] == key) {
                return mid1;
            }
            if (ar[mid2] == key) {
                return mid2;
            }

            if (key < ar[mid1]) {

                r = mid1 - 1;
            }
            else if (key > ar[mid2]) {

                l = mid2 + 1;
            }
            else {

                l = mid1 + 1;
                r = mid2 - 1;
            }
        }

        return -1;
    }
}
```

```

}
public static void main(String args[])
{
    int l, r, p, key;
    int ar[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    l = 0;
    r = 9;
    key = 5;
    p = ternarySearch(l, r, key, ar);
    System.out.println("Index of " + key + " is " + p);
    key = 50;
    p = ternarySearch(l, r, key, ar);
    System.out.println("Index of " + key + " is " + p);
}
}

```

Output:

The screenshot shows an IDE window with a Java file named `GFG.java`. The code defines a `ternarySearch` method that recursively searches for a key in a sorted array. The `main` method calls this function twice: first with `key = 5` and then with `key = 50`. The output console shows the results: "Index of 5 is 4" and "Index of 50 is -1".

```

1  class GFG {
2
3
4  static int ternarySearch(int l, int r, int key, int ar[]) {
5
6      {
7          while (r >= l) {
8
9              int mid1 = l + (r - l) / 3;
10             int mid2 = r - (r - l) / 3;
11
12             if (ar[mid1] == key) {
13                 return mid1;
14             }
15             if (ar[mid2] == key) {
16                 return mid2;
17             }
18
19             if (key < ar[mid1]) {
20

```

Run GFG x

C:\Users\sanja\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=56909:C:\Users\sanja\AppData\Local\Je

Index of 5 is 4

Index of 50 is -1

Process finished with exit code 0

TRIPLET > src > GFG > ternarySearch

4:62 LF UTF-8 4 spaces

26°C Partly cloudy

Search

ENG IN

22:57 21-11-2024

10.Interpolation Search

Time Complexity : $O(\log_2(\log_2 n))$

Solution:

```
import java.util.*;

class GFG {
    public static int interpolationSearch(int arr[], int lo,
                                         int hi, int x)
    {
        int pos;

        if (lo <= hi && x >= arr[lo] && x <= arr[hi]) {

            pos = lo
                + (((hi - lo) / (arr[hi] - arr[lo]))
                  * (x - arr[lo]));

            if (arr[pos] == x)
                return pos;

            if (arr[pos] < x)
                return interpolationSearch(arr, pos + 1, hi,
                                             x);

            if (arr[pos] > x)
                return interpolationSearch(arr, lo, pos - 1,
                                             x);
        }
        return -1;
    }

    public static void main(String[] args)
    {
```

```
int arr[] = { 10, 12, 13, 16, 18, 19, 20, 21,  
            22, 23, 24, 33, 35, 42, 47 };
```

```
int n = arr.length;
```

```
int x = 18;
```

```
int index = interpolationSearch(arr, 0, n - 1, x);
```

```
if (index != -1)
```

```
    System.out.println("Element found at index "  
        + index);
```

```
else
```

```
    System.out.println("Element not found.");
```

```
}
```

```
}
```

Output:

```
1  
2 import java.util.*;  
3  
4 class GFG {  
5  
6     public static int interpolationSearch(int arr[], int lo, 3 usages  
7         int hi, int x)  
8     {  
9         int pos;  
10  
11         if (lo <= hi && x >= arr[lo] && x <= arr[hi])  
12  
13             pos = lo  
14                 + (((hi - lo) / (arr[hi] - arr[lo]))  
15                     * (x - arr[lo]));  
16  
17         if (arr[pos] == x)  
18             return pos;  
19  
20         if (arr[pos] < x)
```

Run GFG x

C:\Users\sanja\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\sanja\AppData\Local\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_rt.jar=57417:C:\Users\sanja\AppData\Local\Je
Element found at index 4

Process finished with exit code 0

TRIPLET > src > @GFG > interpolationSearch

26°C Partly cloudy

Search

ENG IN 23:11 21-11-2024