

Q1. Explain Searching algorithm in data structure?

Ans: Searching is the process of finding a specific elements from the array, collection, List.

Types of Search

1. Linear Search (Sequential Search): Linear Search is the simplest sorting algorithm. It searches for an element by iterating through each element in the array one by one

Algorithm of linear Search

Step1: Start from first element of the Array

Step2: compare each element with the search element

Step3: if the element is found then return its position

Step4: If the loop ends and no match element found then return -1 (element not found)

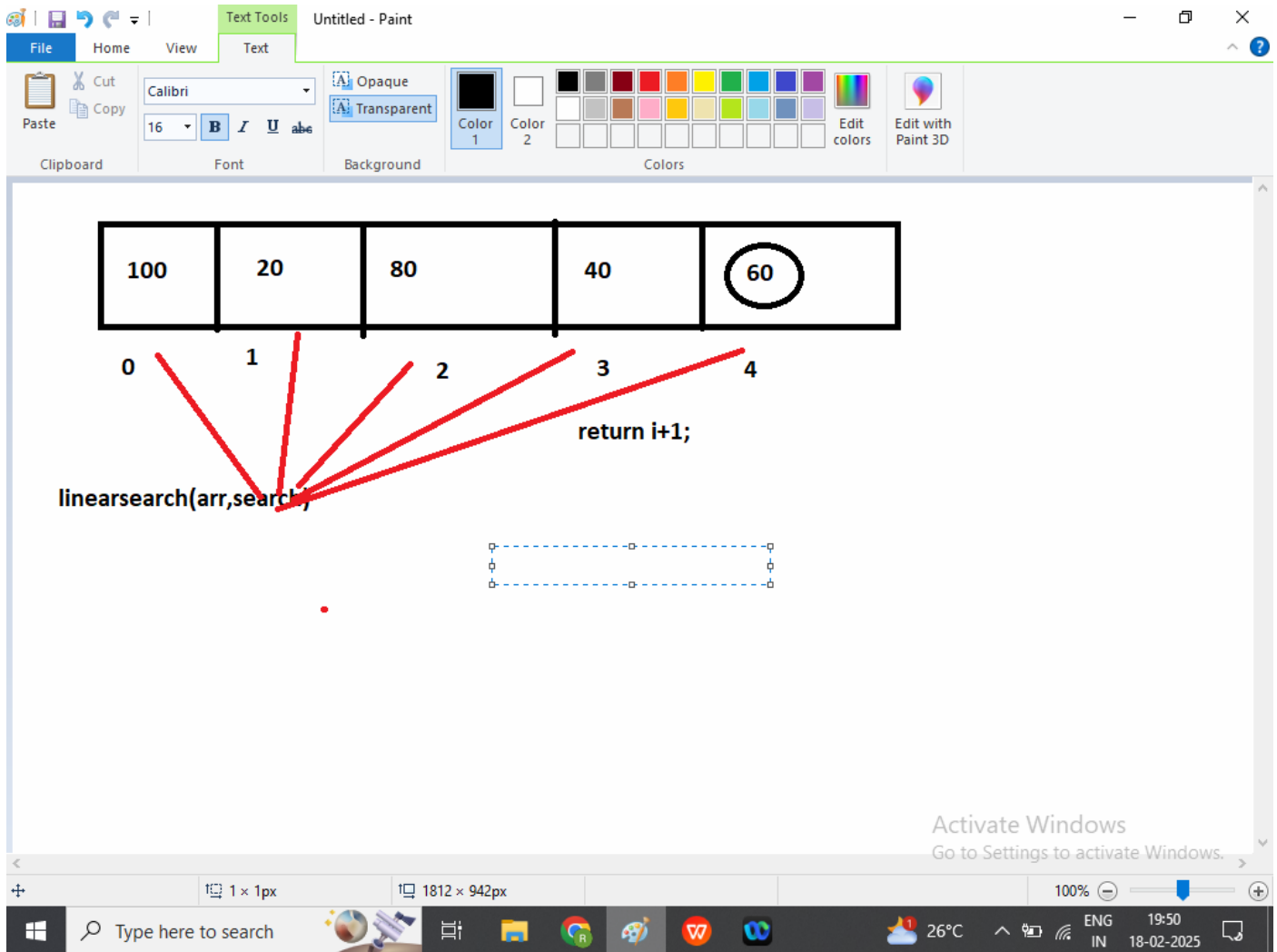
Time Complexity of Linear Search

Best Case : $O(1)$: [found element at first position]

Average Case : $O(n)$

Worst Case: $O(n)$

Space Complexity : $O(1)$



```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dsaj6;

public class LinearSearchDemo {
    public static int linearSearch(int arr[],int search){
        int r=-1;
        for(int i=0;i<arr.length;i++){
            if(arr[i]==search){
                r=i+1;
                break;
            }
        }

        return r;
    }
    public static void main(String[] args) {
        int arr[]={100,20,80,40,60};
        int search=60;
        int result=linearSearch(arr, search);
        if(result>=1){
            System.out.println("Searching Successful : Element found at position "+result);
        }else{
            System.out.println("Searching UnSuccessful : Element Not found");
        }
    }
}

```

```
}  
}
```

2. Binary Search : Binary search is more efficient searching technique but works only on sorted array, it follows divide and conquer strategy

Algorithm of Binary Search

Step1 : Set two pointers with in the method

low=0(start index)

high=arr.length-1(last index)

Step2: Find the middle element

$Mid = (low + high) / 2$

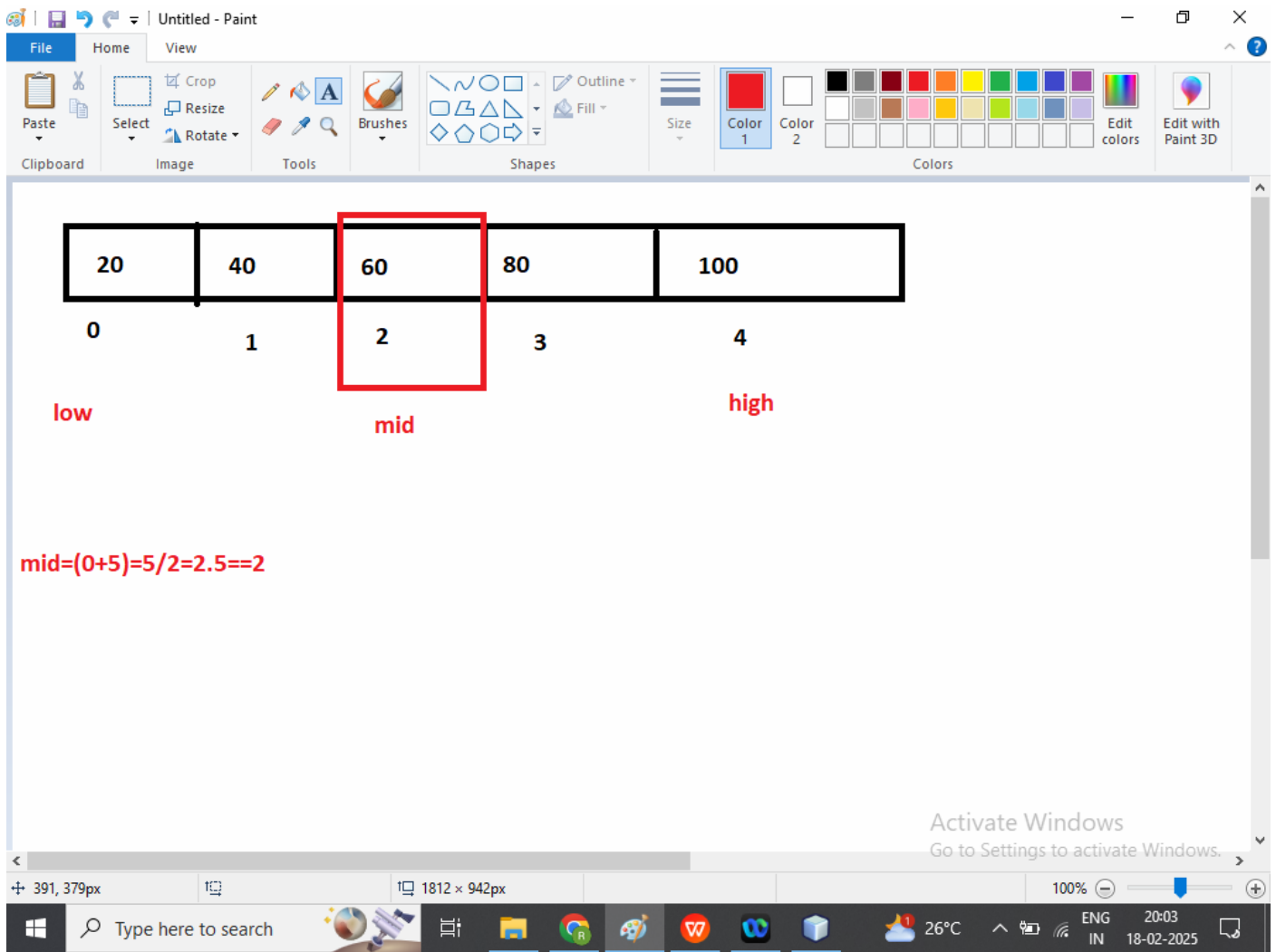
Step3: if arr[mid]=search

Return mid+1;

Stop the loop(break)

If search > mid , search in the right half

If search < mid, search in the left half



Untitled - Paint

File Home View

Paste Select Crop Resize Rotate Image Tools Brushes Shapes Outline Fill Size Color 1 Color 2 Colors Edit colors Edit with Paint 3D

20 40 60 80 100

0 1 2 3 4

high mid-1 low mid+1 high

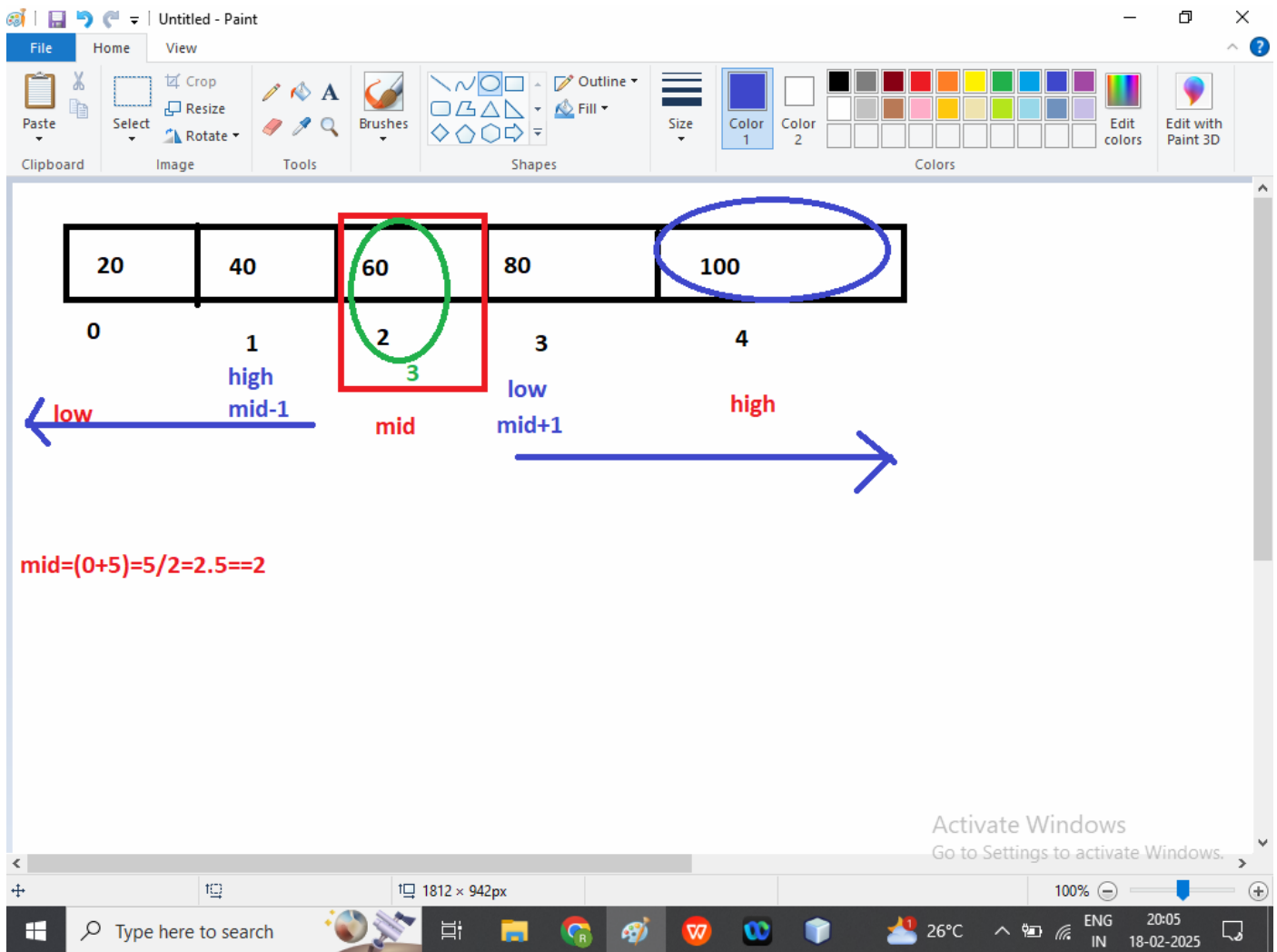
low high

$mid = (0+5)/2 = 2.5 == 2$

Activate Windows
Go to Settings to activate Windows.

386, 303px 1812 x 942px 100%

Type here to search 26°C ENG IN 20:05 18-02-2025



/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

package dsaj6;

```
public class BinarySearchDemo {
    public static int BinarySearch(int arr[],int search){
        int low=0;
        int high=arr.length-1;
        int result=-1;
```

```
while(low<high){  
    int mid=(low+high)/2;  
    if(arr[mid]==search){  
        result=mid+1;  
  
        break;  
    }  
    else if(search>arr[mid]){  
        low=mid+1;  
  
    }else if(search<arr[mid]){  
        high=mid-1;  
    }  
}
```

```
    return result;  
}  
public static void main(String[] args) {  
    int arr[]={20,40,60,80,100};  
    int search=20;  
    int result=BinarySearch(arr,search);  
    if(result>=1){  
        System.out.println("Searching Successful : Element found  
at position "+result);  
    }else{  
        System.out.println("Searching UnSuccessful : Element Not  
found");  
    }  
}
```

}
