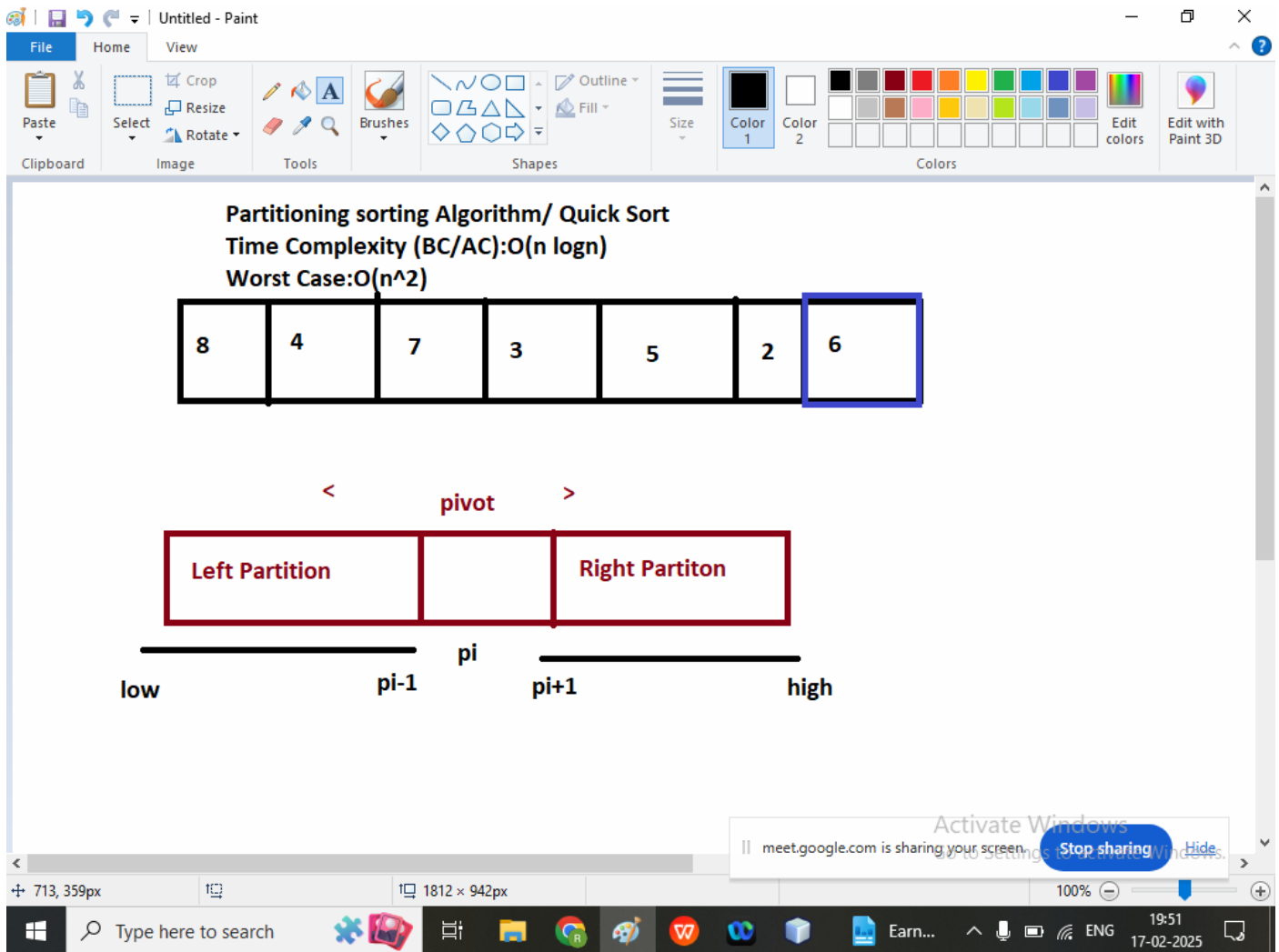


Q1. Explain Quick Sort in data Structure?

Ans: Quick Sort Algorithm is a divide and conquer algorithm that works by selection a pivot element and partitioning the around the pivot such that elements smaller than pivot are on the left and elements are the greater than on the right. The Process is recursively applied on to both partion



Algorithm of Quick Sort

Step1: Select a pivot element(may be first,last,middle element)

Step2: Partition the Array

- => move elements smaller than the pivot to the left
- =>move elements greater than the pivot to the right

Step3: Recursively apply quick sort to the left and right partitions

Step4: Repeat until the entire array is sorted

Example:

Sorting : [8,4,7,3,5,2,6]

Step1: Select pivot element

[8,4,7,3,5,2,6]

Step2: Partitioning

[4,3,5,2,6,8,7]

Step3: Recursively apply Quick Sort

Left Partiton : [4,3,5,2]

Right Partition: [8,7]

Sorting Left Partition [4,3,5,2]

Select pivot element

[4,3,5,2]

Step2: Partition

Left Partiton []

Right Partition [4,3,5]

Select Pivot element

Left Partition: 4,3

Right Partition: []

Left Partition: [4,3]

Left Partition : []

Right Partition [3,4]

Right Partition [8,7]

Pivot element : 7

[7,8]

Final Sorted array

[2,3,4,5,6,7,8]

FileHomeView

Paste

Select

Crop

Resize

Rotate

Image

Tools

Brushes

Shapes

Outline

Fill

Size

Color 1

Color 2

Colors

Edit colors

Edit with Paint 3D

Partitioning sorting Algorithm/ Quick Sort

Time Complexity (BC/AC): $O(n \log n)$

Worst Case: $O(n^2)$

8	4	7	3	5	2	6
---	---	---	---	---	---	---

< pivot >

Left Partiton		Right Partiton
---------------	--	----------------

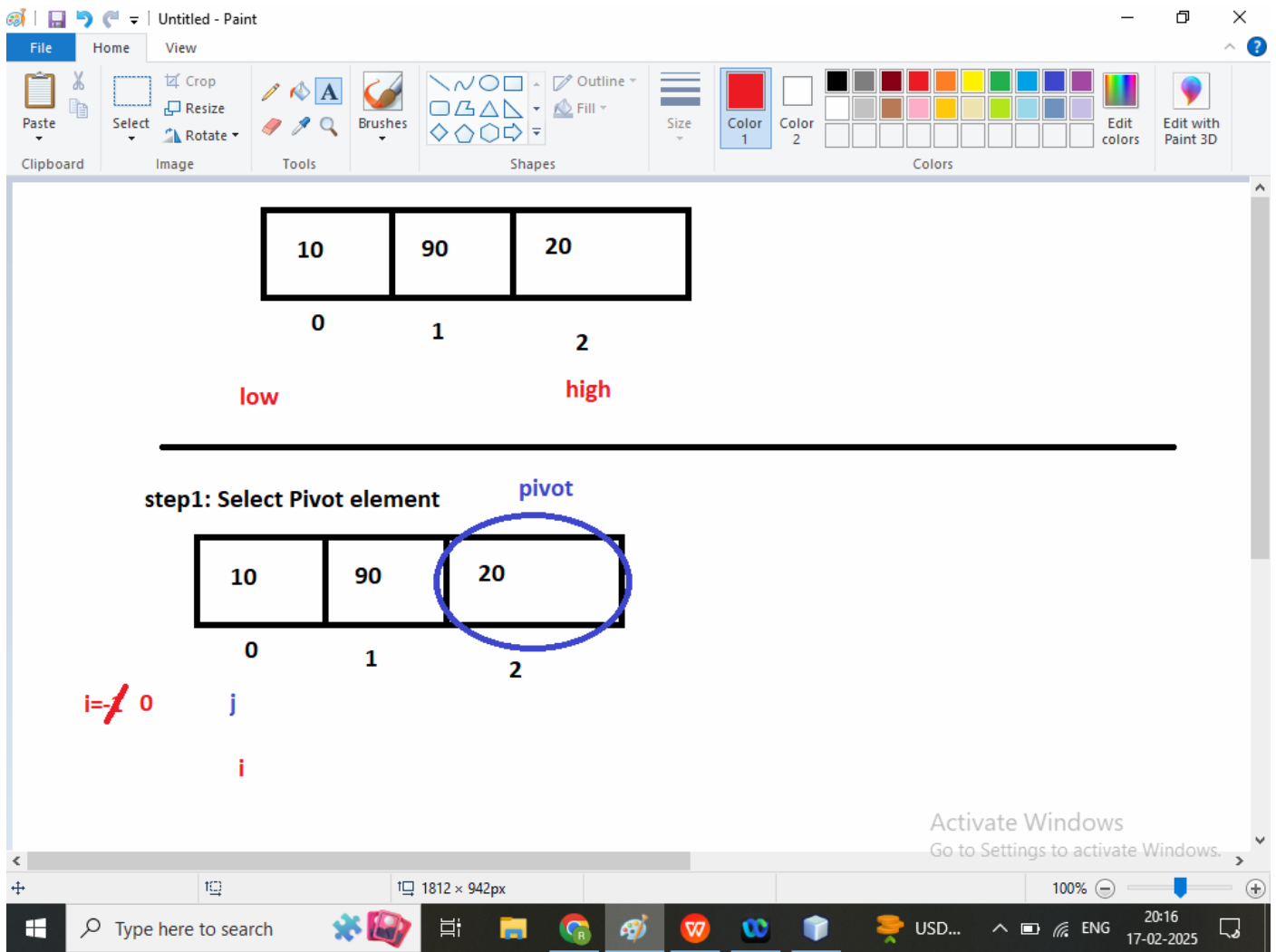
low pi-1 pi pi+1 high

Activate Windows
Go to Settings to activate Windows.

1812 × 942px

100%

20:11
17-02-2025



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

10 90 20

0 1 2

low high

step1: Select Pivot element

10 90 20

0 1 2

~~i = 0~~ i

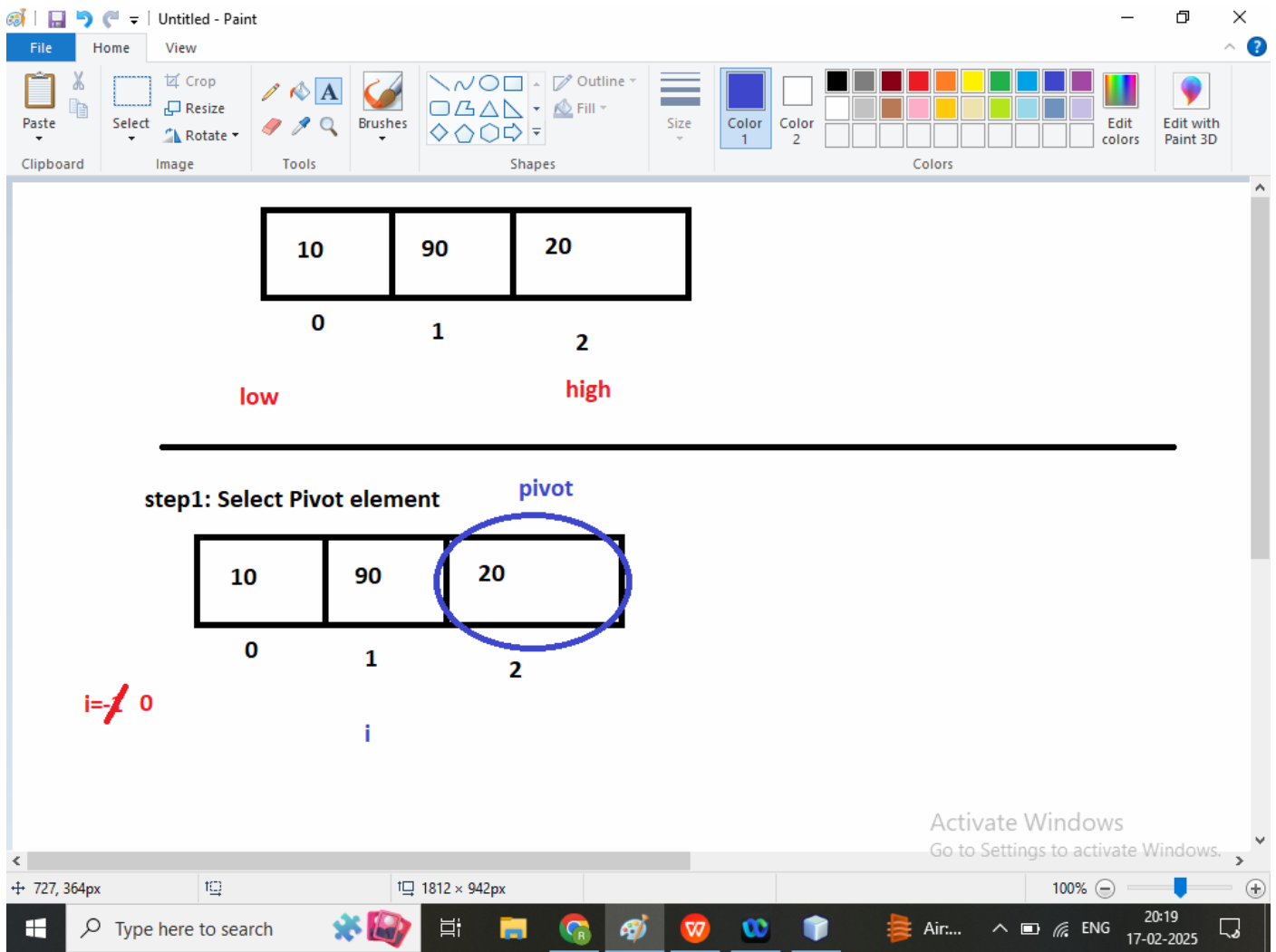
j

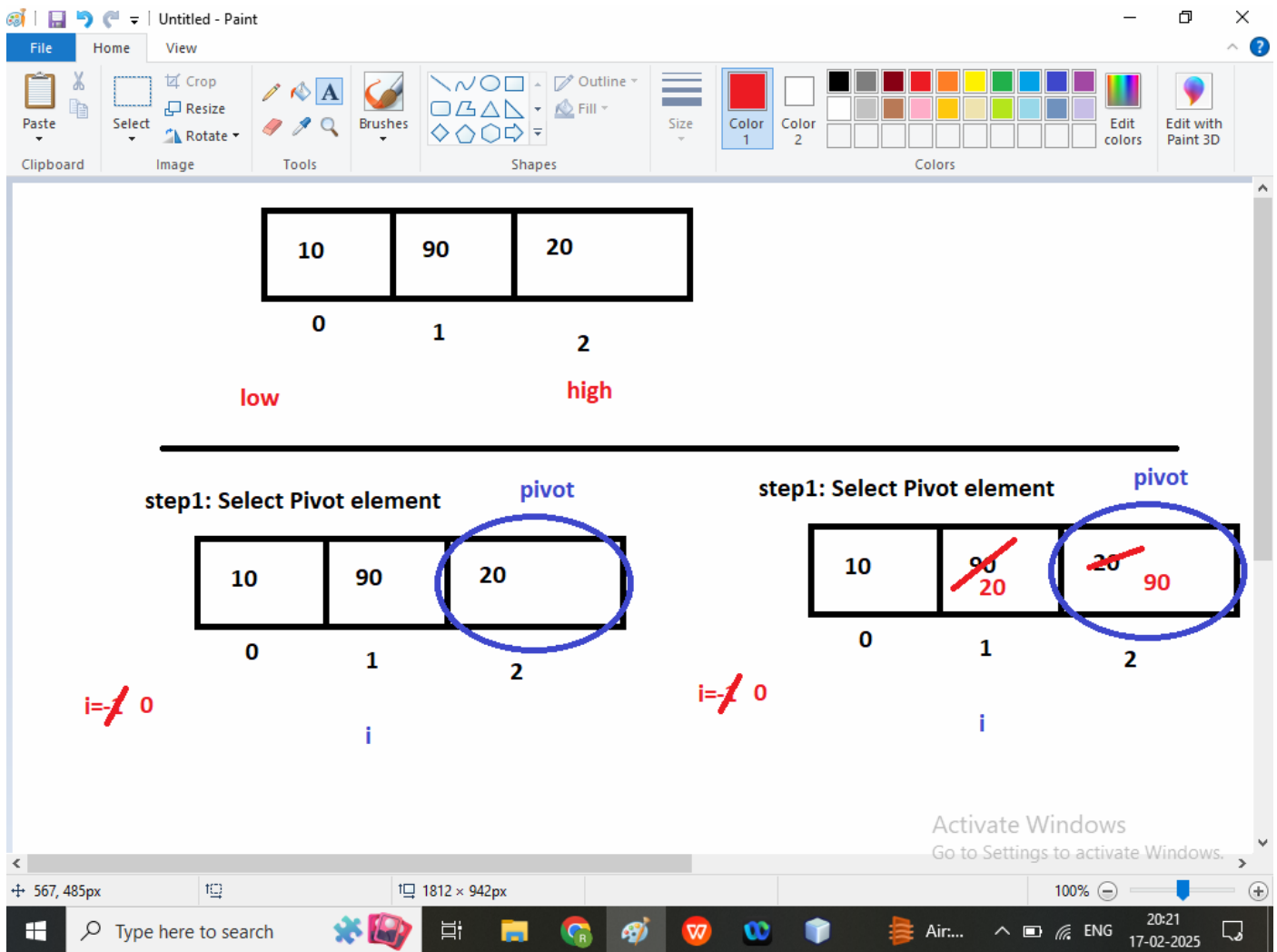
pivot

Activate Windows
Go to Settings to activate Windows.

271,436px 1812 x 942px 100%

Windows Type here to search Air... 20:18 17-02-2025





```

/*
 * 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;

```

```

import com.sun.org.apache.xml.internal.dtm.Axis;

```

```

public class QuickSortDemo {
    public static int partition(int arr[],int low,int high)
    {
        //select pivot element 20
        int pivot=arr[high];
        int i=low-1;//i=-1
        for(int j=low;j<high;j++){//j=2
            if(arr[j]<pivot){
                i++; //0
                int temp=arr[i];//3
                arr[i]=arr[j];//3
                arr[j]=temp;//3
            }
            System.out.println("Partition Method is Called");
        }
        i++;
        //swap with pivot element
        int temp=arr[i];//90

```

```

        arr[i]=arr[high];//20
        arr[high]=temp;//6
        return i;
    }

    public static void quickSort(int arr[],int low,int high)
    {
        if(low<high){
            int pi=partition(arr,low,high);
            //pi=0
            quickSort(arr, low, pi-1);//
            quickSort(arr, pi+1, high);
            // System.out.println("Quick Sort Method is Called");

        }
    }
}

public static void main(String[] args) {
    int arr[]={10,90,20};
    System.out.println("Print Before sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
    quickSort(arr, 0,arr.length-1);
    //quicksort({10,90,20},low=0,high=2)
    System.out.println("\nPrint Before sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
}
}
}

```

```

/*
 * 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;

```

```

import com.sun.org.apache.xml.internal.dtm.Axis;

```

```

public class QuickSortDemo {
    public static int partition(int arr[],int low,int high)
    {
        //select pivot element 20
        int pivot=arr[high];
        int i=low-1;//i=-1
        for(int j=low;j<high;j++){//j=2
            if(arr[j]>pivot){
                i++; //0
                int temp=arr[i];//3
                arr[i]=arr[j];//3
                arr[j]=temp;//3
            }
            System.out.println("Partition Method is Called");
        }
        i++;
        //swap with pivot element
        int temp=arr[i];//90
        arr[i]=arr[high];//20
        arr[high]=temp;//6
        return i;
    }
}

```



```

    }
    public static void quickSort(int arr[],int low,int high)
    {
        if(low<high){
            int pi=partition(arr,low,high);
            //pi=0
            quickSort(arr, low, pi-1);//
            quickSort(arr, pi+1, high);
            // System.out.println("Quick Sort Method is Called");

        }
    }
}
public static void main(String[] args) {
    int arr[]={10,90,20};
    System.out.println("Print Before sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
    quickSort(arr, 0,arr.length-1);
    //quicksort({10,90,20},low=0,high=2)
    System.out.println("\nPrint Before sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
}
}

```

1. Linear Search:
2. Binary Search: