Practical -4(1)

Aim: Write a program to search given element from an array using binary search. Analyze the time complexity for best, average and worst case.

Code:

```
class search
{
   void s(int a[],int k, int c, int low, int high)
   {
      int mid=0;
      int temp=0,temp1=0;
      do
      {
         c++;
         mid=(high+low)/2;
         temp=high;
         temp1=low;
         if(a[mid]==k)
         {
            System.out.println("Key "+k+ " is found at position:  "+mid);
            break;
         }
         else if(k<a[mid])
         {
            high=mid-1;
         }
         else if(k>a[mid])
         {
            low=mid+1;
         }

         //terminating condition:- if low and high both are same after checking above condition
         //it will make loop run infinitly so if they are not changed then loop will terminate
         if(temp==high && temp1==low)
         {
            System.out.println("Key "+ k +" not found!!!");
            break;
         }
      }while(true);
      System.out.println("Total loops run:- "+ c);
   }
}

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

```
    {

        search s=new search();
        int a[]=new int [100];
        for(int i=0;i<100;i++)
        {
           a[i]=i+1;
        }

        int c1=0,c2=0,c3=0,c4=0;
        //sorting
        for(int i=0;i<100;i++)
        {
           c1++;
           c2++;
           c3++;
           c4++;
           for(int j=i+1;j<100;j++)
           {
              c1++;
              c2++;
              c3++;
              c4++;
              if(a[i]>a[j])
              {
                 int temp=a[i];
                 a[i]=a[j];
                 a[j]=temp;
              }
           }
        }


        // Searching

        int low=0,high=99;
        int low1=0,high1=99;
        int low2=0,high2=99;
        int low3=0,high3=99;
        int k1=1,k2=50,k3=100,k4=101;

        //k1
        s.s(a,k1,c1,low,high);
        //k2
        s.s(a,k2,c2,low1,high1);
        //k3
        s.s(a,k3,c3,low2,high2);
        //k4
        s.s(a,k4,c4,low3,high3);
    }
}
```

OUTPUT:

```
PS C:\VSCode\DAA> cd "c:\VSCode\DAA\" ; if ($?) { javac tempbinary.java } ; if ($?) { java tempbinary }
Key 1 is found at position:  0
Total loops run:- 5056
Key 50 is found at position:  49
Total loops run:- 5051
Key 100 is found at position:  99
Total loops run:- 5057
Key 101 not found!!!
Total loops run:- 5058
PS C:\VSCode\DAA>
```

Practical -4(2)

Aim: Write a program to search given element from an array using sequential search . Analyze the time complexity for best, average and worst case.

Code:

```
class search
{
   void time(int a[],int key)
   {
      long startTime1 = System.nanoTime();
      for(int i=0;i<100;i++)
      {
         if(a[i]==key)
         {
            System.out.println("Key "+key+" is found at position: "+i);
            break;
         }
      }
      long endTime1 = System.nanoTime();
      long totalTime1 = endTime1 - startTime1;
      System.out.println("Run-time Is : "+totalTime1+" nanoseconds");
   }
}
class Linear
{
   public static void main(String args[])
   {

      search s=new search();
      int a[] = new int[100];
      for(int i=0;i<100;i++)
      {
         a[i]=i+1;
      }
      int k1=1,k2=50,k3=100;

      s.time(a,k1);
      s.time(a,k2);
      s.time(a,k3);
   }
}
```

OUTPUT:

```
PS C:\VSCode\DAA> cd "c:\VSCode\DAA\" ; if ($?) { javac Linear.java } ; if ($?) { java Linear }
Key 1 is found at position: 0
Run-time Is : 28759500 nanoseconds
Key 50 is found at position: 49
Run-time Is : 120800 nanoseconds
Key 100 is found at position: 99
Run-time Is : 131100 nanoseconds
PS C:\VSCode\DAA>
```