Design and Analysis of Algorithm

Lab10

Name:V.Prasanna Vyshnavi

Regno:19BCE7661

Nearest Neighbour:

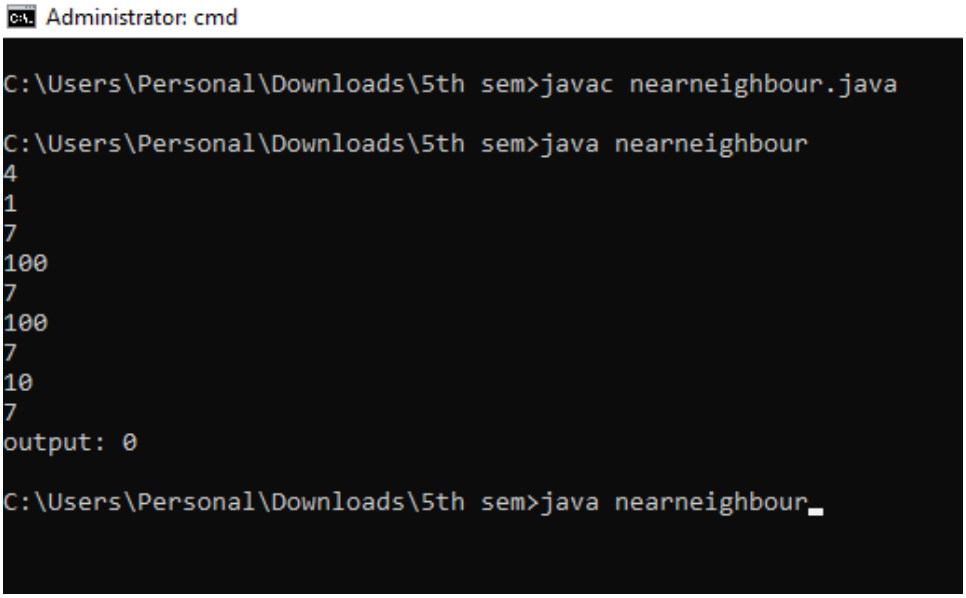
Code:

```
import java.util.*;
public class nearneighbour{
public static void main(String args[]){
Scanner sc=new Scanner(System.in);
int n=sc.nextInt();
int s[]=new int[n];
int e[]=new int[n];
for(int i=0;i<n;i++)
{
s[i]=sc.nextInt();
}
for(int i=0;i<n;i++)
{
e[i]=sc.nextInt();
}
nn(s,e,n);
}

static void nn(int s[],int e[],int n)
```

```
{
int count;
int minlef=Min(s,0,n);
int minrig=Min(e,0,n);
int mind=Math.min(minlef,minrig);
for(int i=0;i<n;i++)
{
for(int j=0;j<n;j++)
{
if(s[i]==e[j])
{
count=0;
}
}
}
System.out.println("output: 0");
}
static int Min(int a[], int index, int l)
{
   int min;
   if (index >= l - 2)
   {
      if (a[index] < a[index + 1])
         return a[index];
      else
         return a[index + 1];
   }
```

```
    min = Min(a, index + 1, l);


    if (a[index] < min)
        return a[index];
    else
        return min;
}
```

Output:



Asymptotic Analysis:

Nearest Neighbor:-

```
int m( s[], e[], n)
  Sort(s[]);
  Sort(e[]);
  minleft = s[0];
  min right = e[0];
  min dis = (s[0], e[0]);
  for i 1 to n:
  while ( s[i] - e[i] >= d) {
        delete e[i], s[i]
  }
  for (i 1 to n):
  {
    best_dist = min (dis(s[i], e[i])
  }
  return best_dist;
```

$$T(n) = T(n/2) + T(n/2) + c \cdot n$$

$$= 2\left[2T(n/2) + c(n/2)\right] + c \cdot n$$

$$= 2^2 T(n/2^2) + 2c \cdot n$$

⋮ K times

$$T(n) = 2^K T(n/2^K) + k \cdot c \cdot n$$

$$= 2^K T(1) + \log_2^n \cdot c \cdot n$$

$$= 0 + \log_2^n \cdot c \cdot n$$

$$T(n) = O(n \log_2^n)$$