 *DEPARTMENT OF INFORMATION TECHNOLOGY*

Experiment No.4

|  |  |  |
| --- | --- | --- |
| **Experiment Number** | **4** | |
| **Experiment Title** | **Bubble Sort (Ascending) and Linear Search** | |
| **Resources / Apparatus Required** | Java Developer Kit , Command Prompt | Standard PC with Windows 7,8 or 10 |
| **Objectives**  **(Skill Set / Knowledge Tested / Imparted)** | To Learn the concept of sorting and searching within an array. | |
| **Theory** | **Bubble sort**, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems even when compared to insertion sort It can be practical if the input is usually in sorted order but may occasionally have some out-of-order elements nearly in position.  **Linear search** or sequential search is a method for finding a target value within a list. It sequentially checks each element of the list for the target value until a match is found or until all the elements have been searched. Linear search runs in at worst linear time and makes at most n comparisons, where n is the length of the list. Linear search is rarely practical because other search algorithms and schemes, such as the binary search algorithm and hash tables, allow significantly faster searching for all but short lists. | |
| **Program & output** | import java.lang.\* ;  import java.util.\* ;  class AscSort  {  public static void main(String args[])  {  int n,i,j,t ;  Scanner s=new Scanner(System.in) ;  System.out.print("\nEnter Number of Elements in Array ") ;  n=s.nextInt() ;  int[] a = new int[n] ;    System.out.print("\nEnter the Elements\n\n") ;  for(i=0;i<n;i++)  {  a[i]=s.nextInt() ;  }    for(i=0;i<n;i++)  {  for(j=0;java<n-1;j++)  {  if(a[j]>a[j+1])  {  t=a[j] ;  a[j]=a[j+1] ;  a[j+1]=t ;  }  }  }    System.out.print("\nThe Sorted Array is ") ;  for(i=0;i<n;i++)  {  System.out.print(a[i]+" ") ;  }      System.out.println() ;  }  }  C:\Users\Ashu\Desktop\Coding\Codes\Java Programs\Screen Shots\AscSort.png  import java.lang.\* ;  import java.util.\* ;  class Search  {  public static void main(String args[])  {  int n,i,x ;  Scanner s=new Scanner(System.in) ;  System.out.print("\nEnter Number of Elements in Array ") ;  n=s.nextInt() ;  int[] a = new int[n] ;    System.out.print("\nEnter the Elements ") ;  for(i=0;i<n;i++)  {  a[i]=s.nextInt() ;  }    System.out.print("\nEnter the Element to be Searched ") ;  x=s.nextInt() ;  for(i=0;i<n;i++)  {  if(a[i]==x)  {  System.out.print("\nElement is at Index location " + i) ;  break ;  }  }    if(i==n)  {  System.out.print("\nElement not Found") ;  }  System.out.println() ;  }  }  C:\Users\Ashu\Desktop\Coding\Codes\Java Programs\Screen Shots\Search.png\ | |
| **Conclusion** | Hence concept of array can be used to sort and search data | |