Linear Search

1. Linear Search is the simplest searching algorithm.

2. It traverses the array sequentially to locate the required element.

3. It searches for an element by comparing it with each element of the array one by one.

4. So, it is also called as Sequential Search.

Algorithm for Linear search:

Consider-

1. There is a linear array ‘a’ of size ‘n’.

2. Linear search algorithm is being used to search an element ‘item’ in this linear array.

3. If search ends in success, it sets loc to the index of the element otherwise it sets loc to -1.

Linear Search Example:

The following linear array.

Element 15 has to be searched in it using Linear Search Algorithm.

Now,

Linear Search algorithm compares element 15 with all the elements of the array one by one.

It continues searching until either the element 15 is found or all the elements are searched.

Step1:

It compares element 15 with the 1st element 92.

Since 15 ≠ 92, so required element is not found.

So, it moves to the next element.

Step2:

It compares element 15 with the 2nd element 87.

Since 15 ≠ 87, so required element is not found.

So, it moves to the next element.

Step3:

It compares element 15 with the 3rd element 53.

Since 15 ≠ 53, so required element is not found.

So, it moves to the next element.

Step4:

It compares element 15 with the 4th element 10.

Since 15 ≠ 10, so required element is not found.

So, it moves to the next element.

Step5:

It compares element 15 with the 5th element 15.

Since 15 = 15, so required element is found.

Now, it stops the comparison and returns index 4 at which element 15 is present.

Binary Search

```
Binary_Search(a, lower_bound, upper_bound, val) // 'a' is the given array, 'lower_bound' is the index of the first array element, 'upper_bound' is the index of the last array element, 'val' is the value to search

Step 1: set beg = lower_bound, end = upper_bound, pos = - 1

Step 2: repeat steps 3 and 4 while beg <=end

Step 3: set mid = (beg + end)/2

Step 4: if a[mid] = val

    set pos = mid

    print pos

    go to step 6
```

```
else if a[mid] > val
set end = mid - 1
else
set beg = mid + 1
[end of if]
[end of loop]
Step 5: if pos = -1
print "value is not present in the array"
[end of if]
Step 6: exit
```

Exercise

Write a Pseudocode for these problems.

```
1.S=(A+B+C)/Y
Step1: start
Step2: Declare the variables for A,B,C,Y
Step3: Get the input values for A,B,C,Y
Step4: Then calculate S
Step5: Then Total = A + B + C
Step 6: Then divide the Total    with Y
                    S = total / Y
Step7: Then Display the result S and print S
Step8: Stop
```

2.Convert from Celcius to Farenheit (Multiply by 9, then divide by 5, then add 32).

Step1: start

Step2: Declare the variable for Celcius

Step3: F=(9/5)*Celcius +32

Step4: Print the Output

Step5: Display the result

Step6: Stop

3.Area of Circle(A=Pie r Square)

Step1: Start

Step2: The user should enter a value for radius

Step3: Calculate Area = pi * radius* radius

Step4: pi=22/7 or 3.14

Step5: Display the area

Step6: Stop

4.Volume of sphere(4/3 Pie r cube)

Step1: Start

Step2: initialize pi value(pi=22/7 or 3.14)

Step3: The user should enter the radius of sphere

Step4: the volume of sphere v=4/3 Pie r cube

Step5: tha volume of sphere V

Step6: Stop

5.Average Speed=Distance Travelled/Time Taken

Step1: Start

Step2: Assign the values for Distance and Time

Step3: calculate its speed

Step4: The formula for Speed is Speed = Distance / Time

Step5: Stop

Insertion Sort Algorithm

----

a sorting algorithm that places an unsorted element at its suitable place in each iteration. Insertion sort works similarly as we sort cards in our hand in a card game. We assume that the first card is already sorted then, we select an unsorted card.

*Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands.

The advantage of insertion sort:

1.The pure simplicity of the algorithm.

2.The ability to sort a list as it is being received.

3.The relative order of items with equal keys does not change.

The Steps of Insertion Sort:

1.Insertion Algorithms: Steps on how it works:

2.If it is the first element, it is already sorted.

3.Pick the next element.

4.Compare with all the elements in sorted sub-list.

5.Shift all the the elements in sorted sub-list that is greater than the value to be sorted.

6.Insert the value.

7.Repeat until list is sorted.

Where is insertion sort used?

used in complex computer programs such as file search, data compression, and path finding.

Insertion Sort Algorithm:

Bubble sort Algorithm

Bubble sort is a basic algorithm for arranging a string of numbers or other elements in the correct order.

the five steps of the bubble sort algorithm?

1.Bubble sort

2.Start at the beginning of the list.

3.Compare the first value in the list with the next one up. If the first value is bigger, swap the positions of the two values.

4.Move to the second value in the list. ...

5.Keep going until there are no more items to compare.

6.Go back to the start of the list.

Case Study Example

1. Invite your frnds for your birthday party at your home . so one of your frnd is on one destination so he needs help to come to our home .how will you guide him.(clue : time and cost).

Cases are:

Case1: once you reach bus stop call me

Case2: once you reach take auto

Case3: once you reach take next bus

Answers:

Case1: The cost will be less and time effeciency will be more

Case2: The cost will be more and time will be less

Case3: The cost will be less and time will be more