## Experiment No. : 8

## Title: Virtual Lab on Searching and Sorting

**Batch: B1** **Roll No.: 1914078** **Experiment No.: 8**

**Aim:** Explore the virtual lab on Sorting and Searching Algorithms

**Resources needed:** Web Browser and flash player plug-in

## Binary Search and Bubble Sort -
## BINARY SEARCH

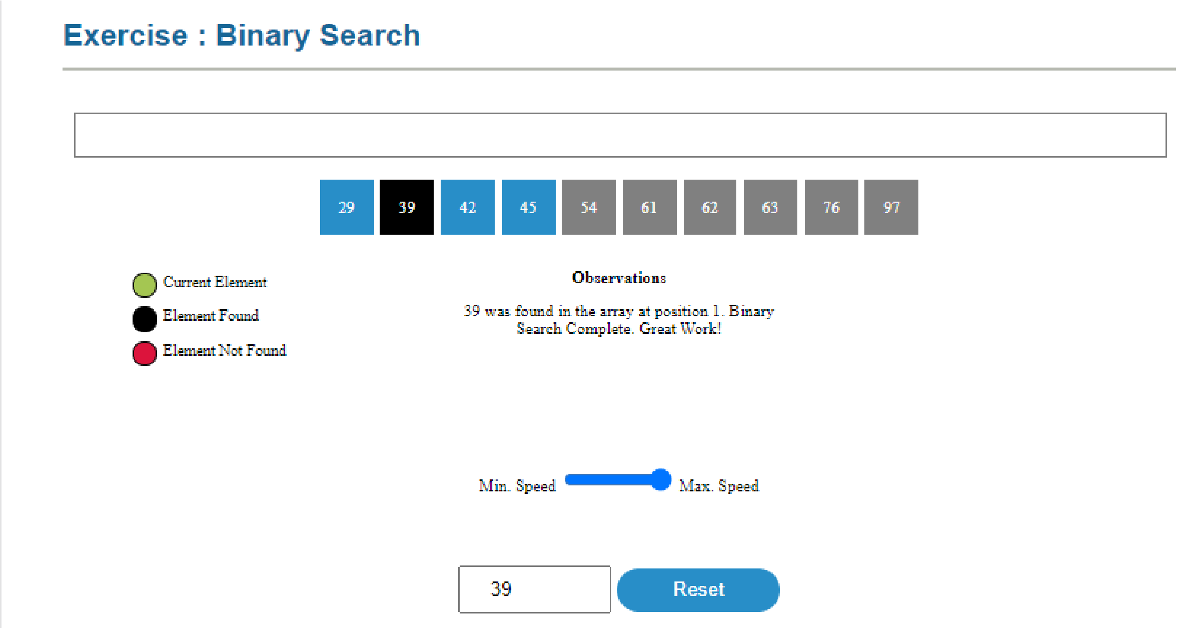
1) Introduction

   Binary search is an efficient algorithm for finding an item from a sorted list of items. It works by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible locations to just one.

2) Objective

   Objective of binary search is to check whether the element given by the user in present in a list and if it is present then at what position. This is achieved by initially finding the middle element of the list, comparing the searched element with it, if greater then the elements greater than the middle terms are checked and if smaller the terms smaller than the middle term are checked. This method is repeated till the term is found and if not found element is not found is displayed.
   The list is always to be arranged in ascending order before starting binary search.

3) Observations from simulation



Exercise : Binary Search

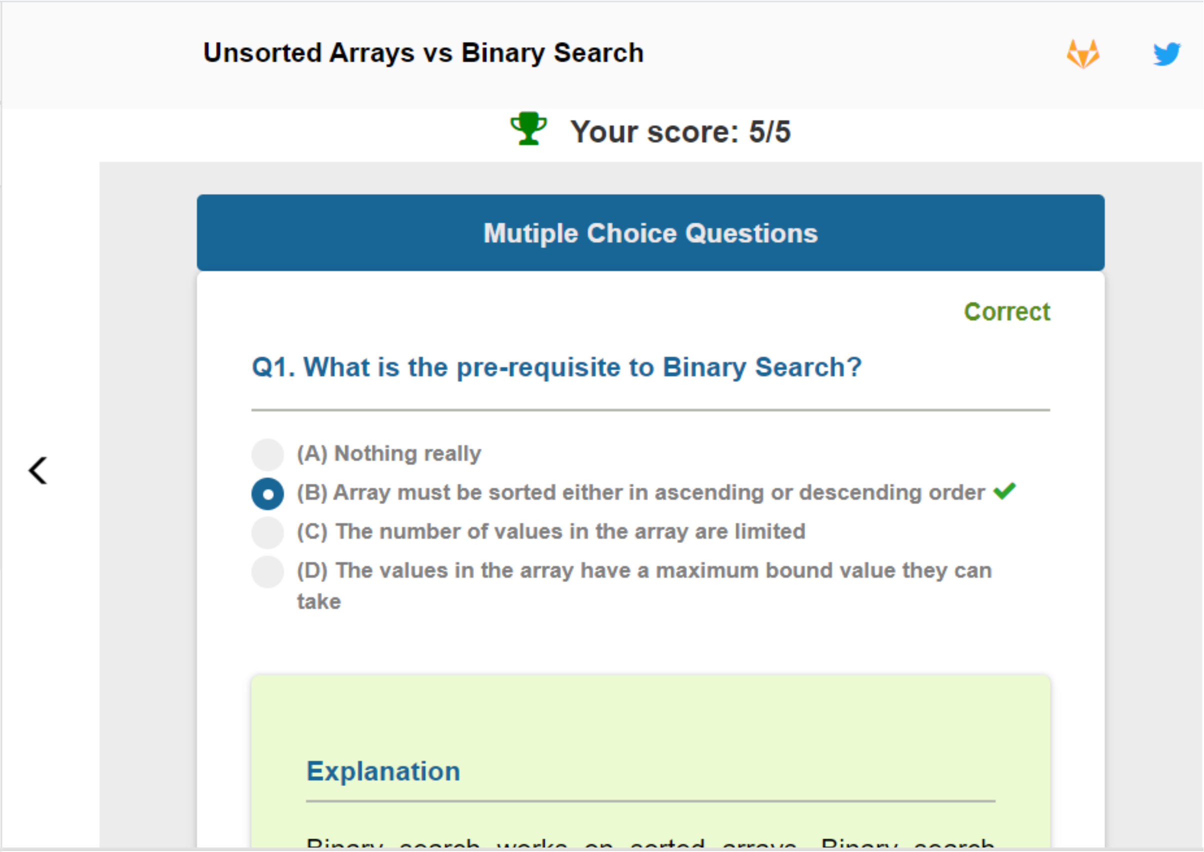
29 39 42 45 54 61 62 63 76 97

Observations

39 was found in the array at position 1. Binary Search Complete. Great Work!

39

4) Questions and Answers



Your score: 5/5

Mutiple Choice Questions

Correct

Q1. What is the pre-requisite to Binary Search?

○ (A) Nothing really

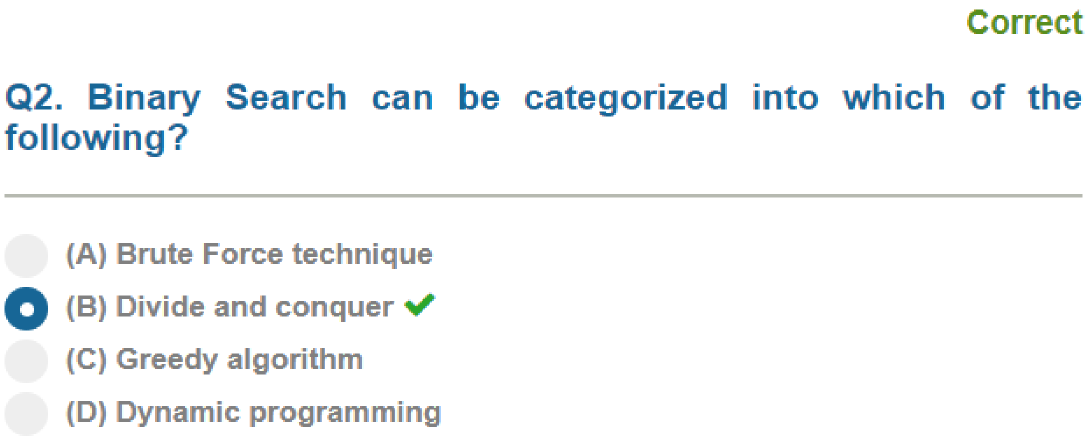
● (B) Array must be sorted either in ascending or descending order ✓

○ (C) The number of values in the array are limited

○ (D) The values in the array have a maximum bound value they can take

Explanation

Binary search works on sorted arrays. Binary search



Correct

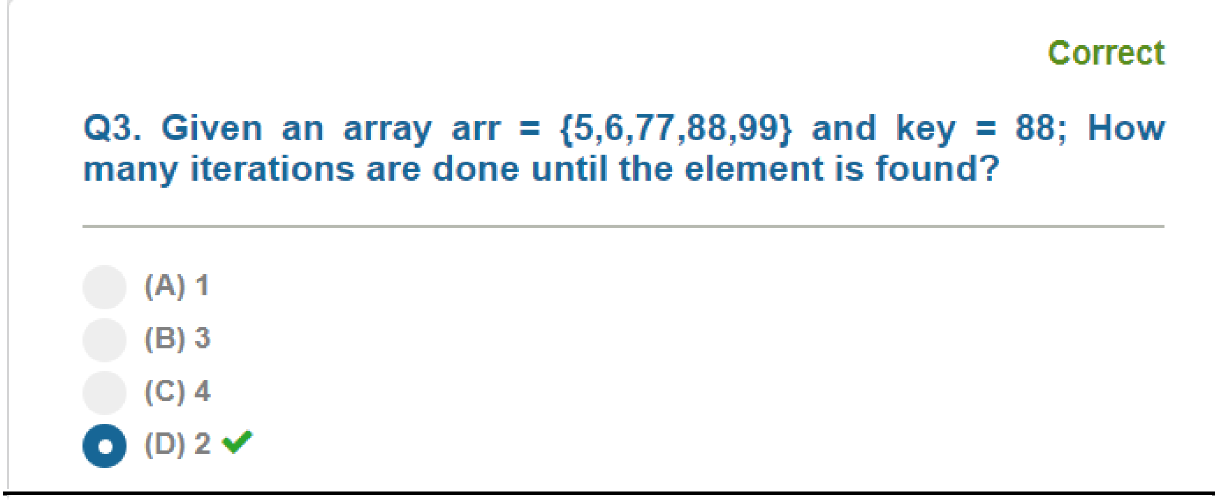
Q2. Binary Search can be categorized into which of the following?

○ (A) Brute Force technique

● (B) Divide and conquer ✓

○ (C) Greedy algorithm

○ (D) Dynamic programming



Correct

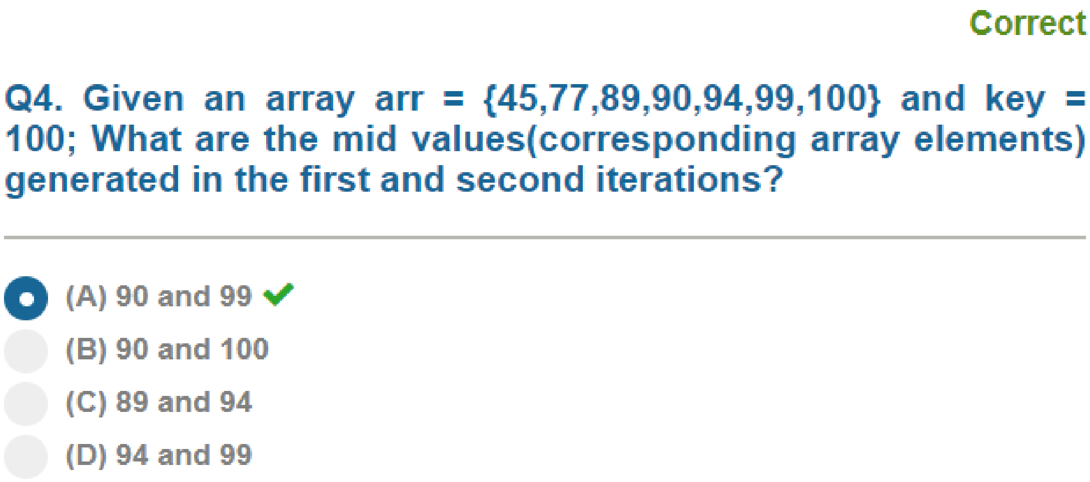
Q3. Given an array arr = {5,6,77,88,99} and key = 88; How many iterations are done until the element is found?

○ (A) 1

○ (B) 3

○ (C) 4

● (D) 2 ✓



Correct

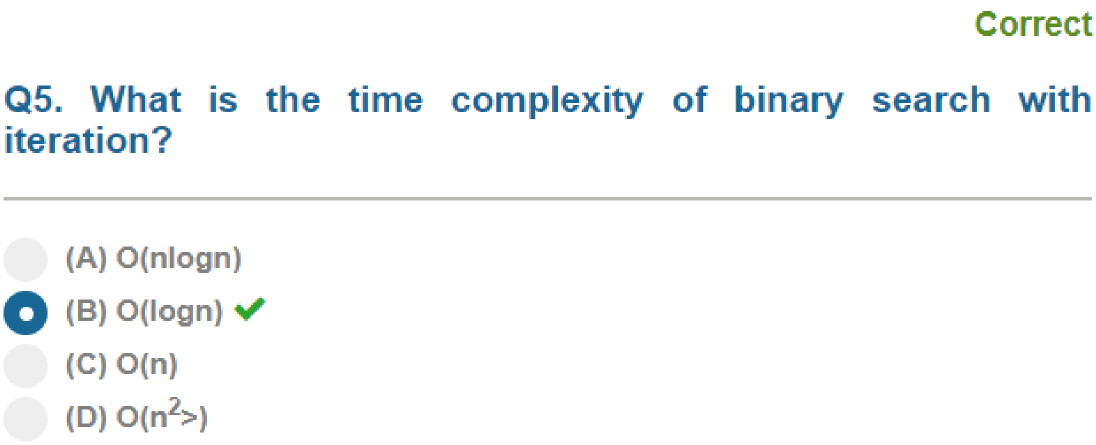
Q4. Given an array arr = {45,77,89,90,94,99,100} and key = 100; What are the mid values(corresponding array elements) generated in the first and second iterations?

● (A) 90 and 99 ✓

○ (B) 90 and 100

○ (C) 89 and 94

○ (D) 94 and 99



Correct

Q5. What is the time complexity of binary search with iteration?

○ (A) O(nlogn)

● (B) O(logn) ✓

○ (C) O(n)

○ (D) O($n^2$>)

## BUBBLE SORT:

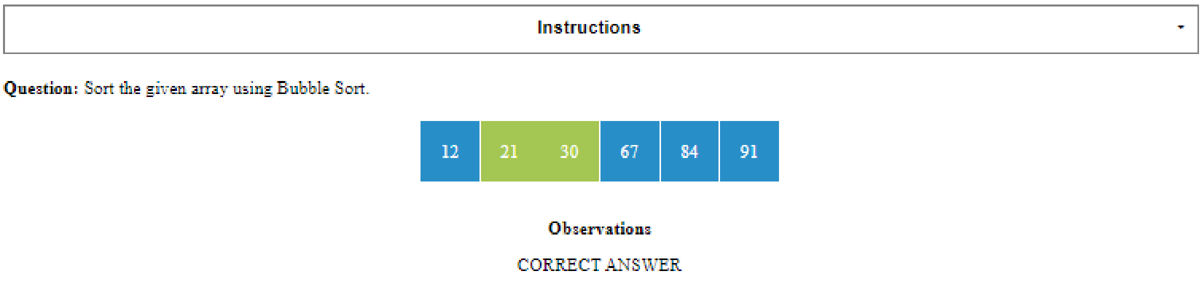
1) Introduction

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted.

2) Objective

Objective of bubble sort is to sort an array or a list in an order. This is achieved by comparing adjacent values is the list and switching their position on the basis of greater element. Then this process is repeated till all the elements are sorted.
An opposite of the method can be done to arrange the list in a descending order as well.

3) Observations from simulation



Instructions

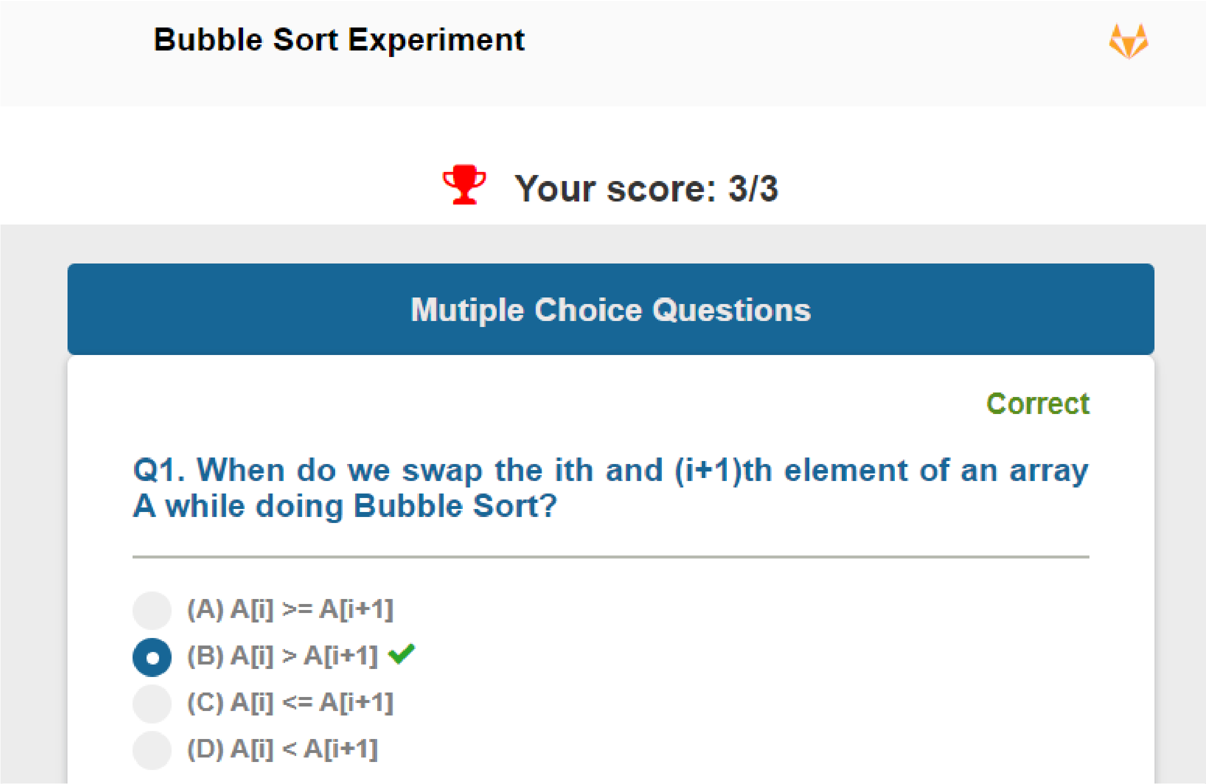
**Question:** Sort the given array using Bubble Sort.

| | | | | | |
|---|---|---|---|---|---|
| 12 | 21 | 30 | 67 | 84 | 91 |

**Observations**

CORRECT ANSWER

4) Questions and Answers



**Bubble Sort Experiment**

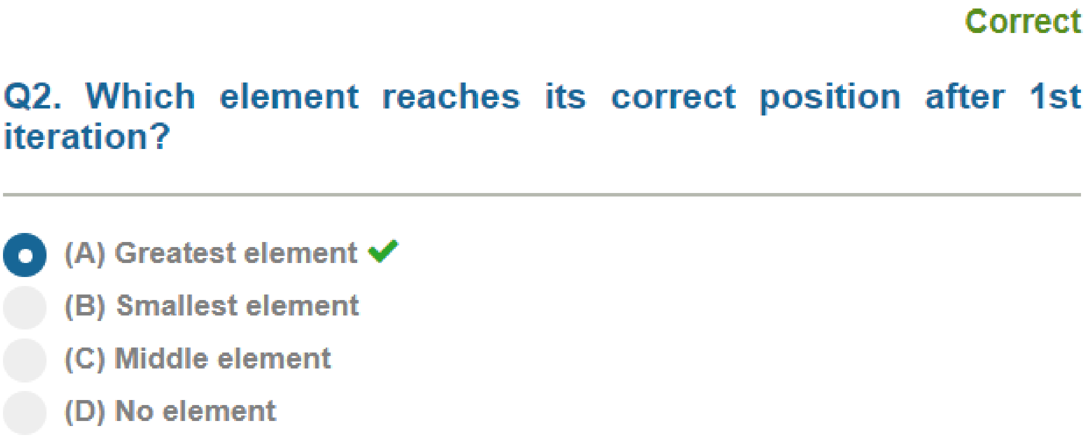
Your score: 3/3

Mutiple Choice Questions

Correct

Q1. When do we swap the ith and (i+1)th element of an array A while doing Bubble Sort?

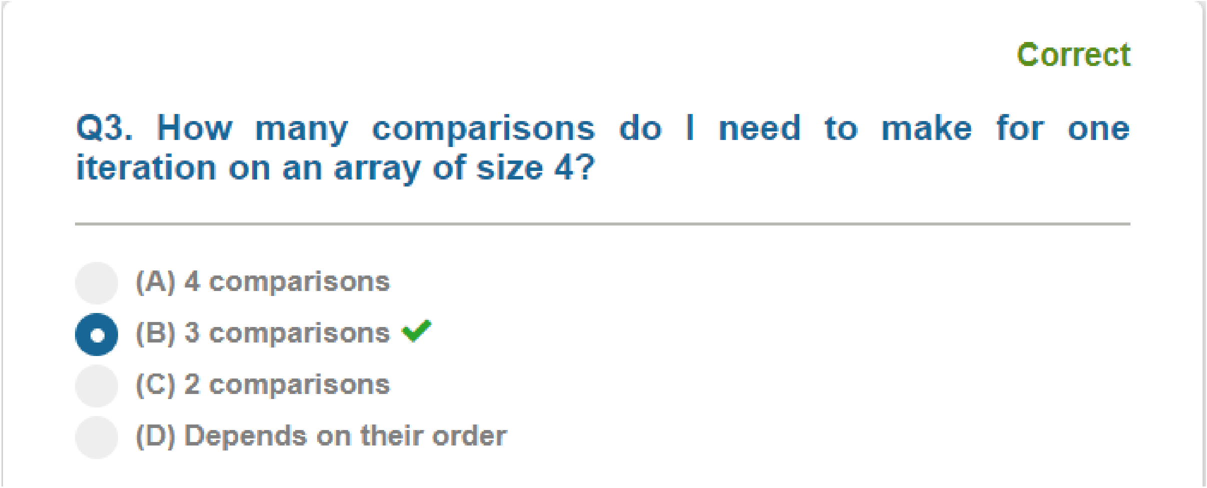
○ (A) A[i] >= A[i+1]
● (B) A[i] > A[i+1] ✓
○ (C) A[i] <= A[i+1]
○ (D) A[i] < A[i+1]



Correct

Q2. Which element reaches its correct position after 1st iteration?

● (A) Greatest element ✓
○ (B) Smallest element
○ (C) Middle element
○ (D) No element



Correct

Q3. How many comparisons do I need to make for one iteration on an array of size 4?

○ (A) 4 comparisons

● (B) 3 comparisons ✓

○ (C) 2 comparisons

○ (D) Depends on their order

**Algorithm: Pseudo code for both**

**BINARY SEARCH:**

**Algorithm:**

STEP 1: Start with the middle element.

- → If the target value is equal to the middle element of the array, then return the index of the middle element.
- → If not, then compare the middle element with the target value,
    1. If the target value is greater than the number in the middle index, then pick the elements to the right of the middle index, and start with Step 1.
    2. If the target value is less than the number in the middle index, then pick the elements to the left of the middle index, and start with Step 1.

STEP 2: When a match is found, return the index of the element matched.

STEP 3: If no match is found, then return failure.

**Pseudo code:**

```
lower=0;

upper=n-1;

while (upper > lower)

  mid = (lower + upper)/2;

       if (a[mid] ==data)

            break;  //exit if found
```

```
          else if (a[mid] > data)
                upper=mid-1;
          else
               lower=mid+1;
end while
if(upper < lower)
    data not found
else
    data found at index mid
```

**BUBBLE SORT:**

**Algorithm:**

STEP 1: Compare the $i^{th}$ and $(i+1)^{th}$ element, where i=first index to i=second last index.

STEP 2: Compare the pair of adjacent elements. If $i^{th}$ element is greater than the $(i+1)^{th}$ element, swap them.

STEP 3: Run steps 1 and 2 a total of N-1 times to attain the final sorted array.

**Pseudo code:**

```
for i=0 to size-1
      for j=0 to size-i-1
          if  arr[j] > arr[j+1]
               swap (arr[j], arr[j+1])
         end if
    end inner for loop
end outer for loop
```

**Activity:** Students are expected to go through virtual lab/visualizations, study it and write some primary observation. Complete the writeup by writing pseudo codes and working examples and finally write the conclusion.

**Outcomes:**

**CO4: Demonstrate different sorting and searching methods.**

**Conclusion:**

Successfully understood the concept of binary search and bubble sort by using virtual lab.

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date**

**References:**

**Books/ Journals/ Websites:**

- Y. Langsam, M. Augenstin and A. Tannenbaum, “Data Structures using C”, Pearson Education Asia, 1st Edition, 2002
- https://ds1-iiith.vlabs.ac.in/exp/bubble-sort/exp.html#Bubble%20Sort%20Experiment
- https://ds1-iiith.vlabs.ac.in/exp/unsorted-arrays/exp.html#Binary%20Search
- https://www.cs.usfca.edu/~galles/visualization/Search.html
- https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/implementing-binary-search-of-an-array