Module Code: CSE401

Session 10: Search & Sort Methods

Session Speaker:

Prema Monish
premamonish.tlll@msruas.ac.in

Session Objectives

- At the end of this lecture, student will be able to
  - apply the general strategies of searching in a list of elements
  - use various approaches of searching in a list of elements
  - apply the general strategies of sorting in a list of elements
  - identify the various requirements that demand different strategies of sorting

Contents

- Linear Search
- Binary Search
- Bubble sort
- Selection Sort

Searching

- Searching
  - The process of finding a particular element (Key value) of an array

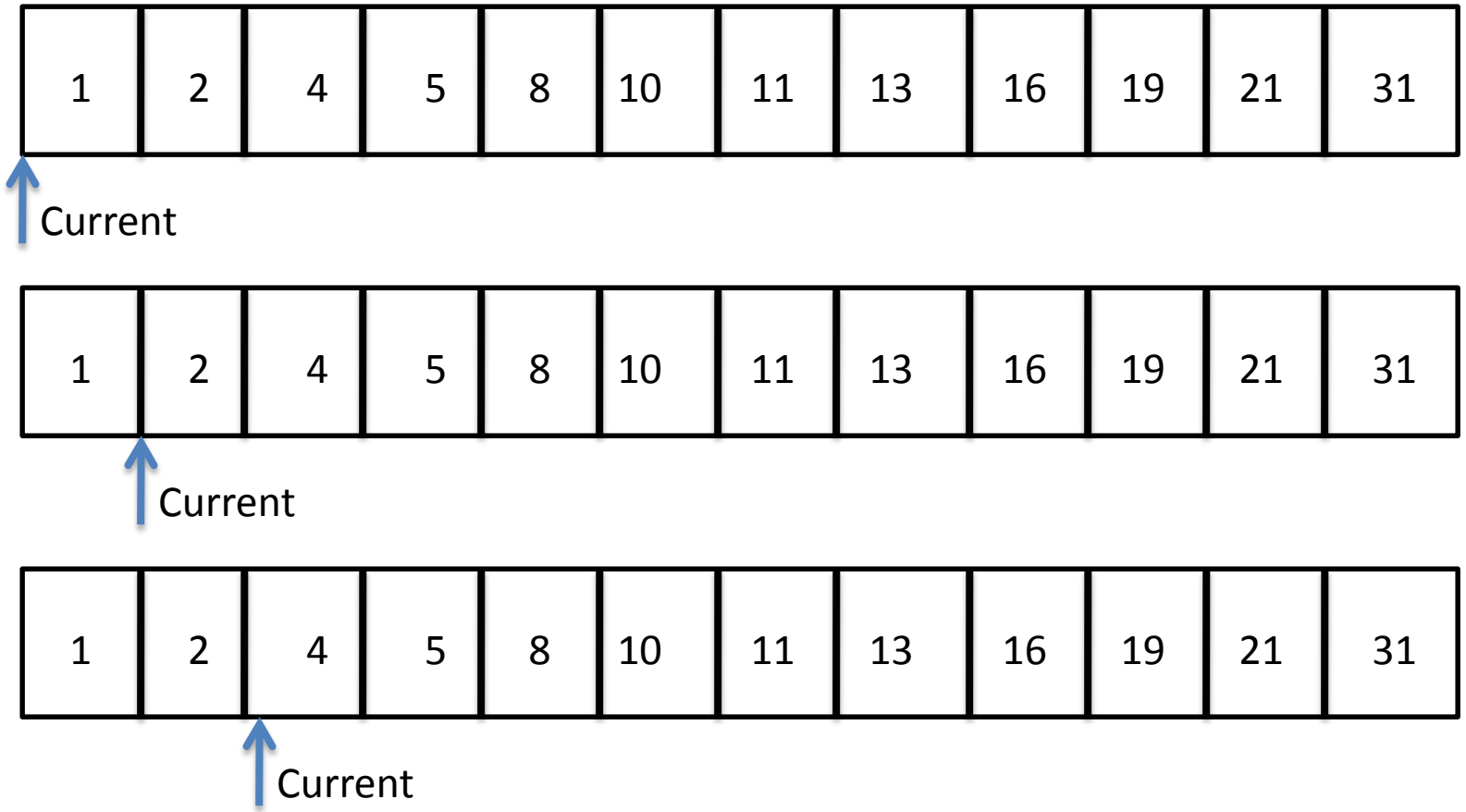
- Two types
  1. Linear search
  2. Binary search

Linear Search

- Also known as sequential search
- Suitable for searching a set of data for a particular value

- It operates by checking every element of a list one at a time in sequence until
  - a match is found or
  - end of array is reached

Trace of Linear Search



Merits and Demerits of Linear Search

- Works well for small or unsorted arrays
- Easy to implement

- Inefficient for large arrays

Binary Search

- Assumes that the data in the array is ***Sorted***
- A technique for finding a particular value in a linear array, by ruling out half of the data at each step

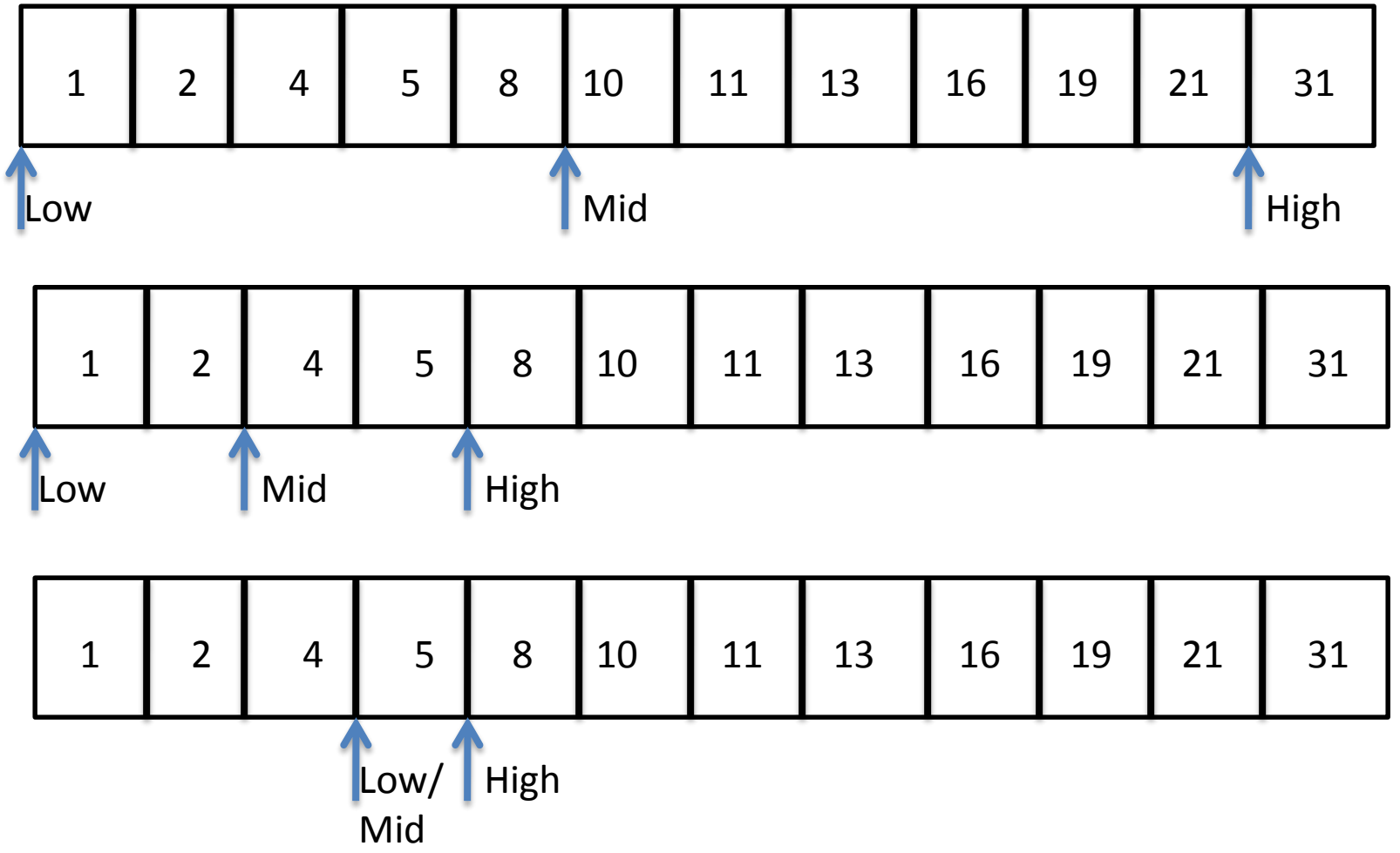
Binary Search Algorithm

- Locate the middle element of the array and compares it to the search key

- If they are equal
  - search key is found
- If they are not equal
  - problem is reduced to searching one-half of the array

- If the search key is less than the middle element of the array
  - the first half of the array is searched
  - otherwise the second half of the array is searched

Trace of Binary Search

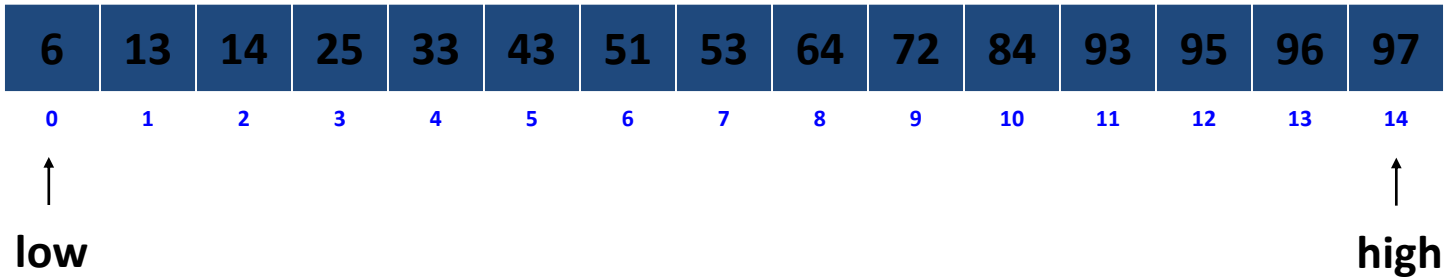


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

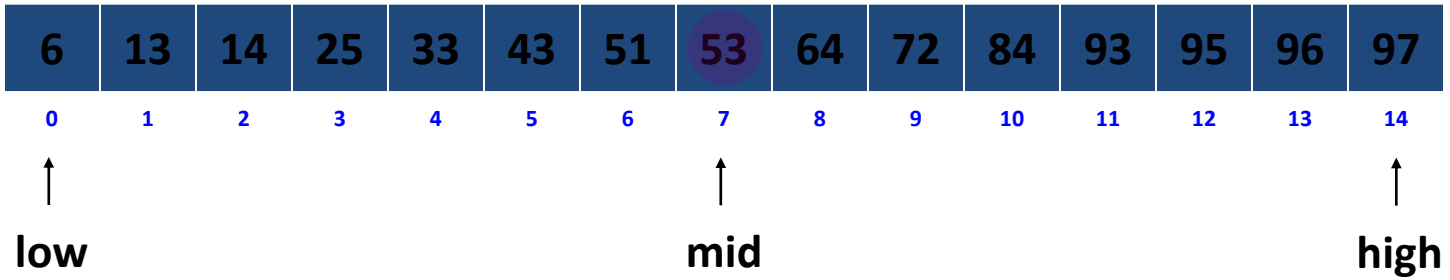


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

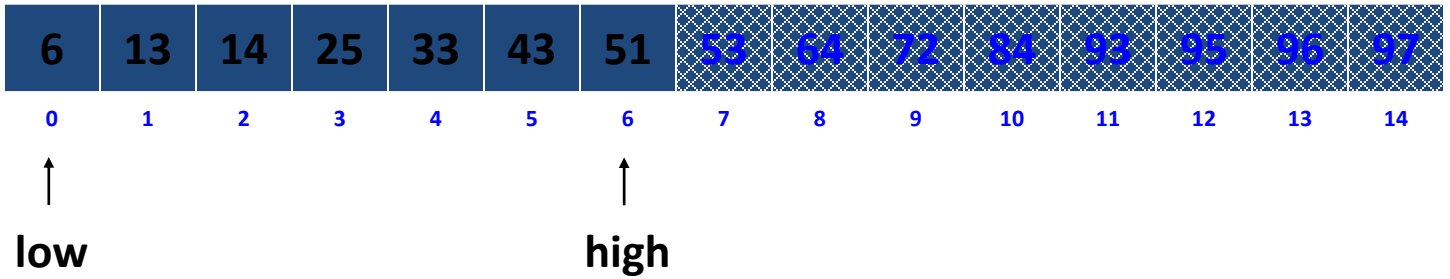


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

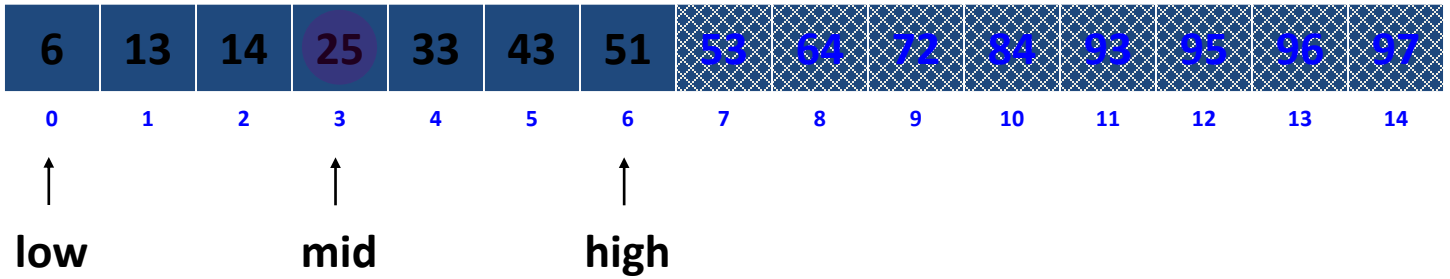


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

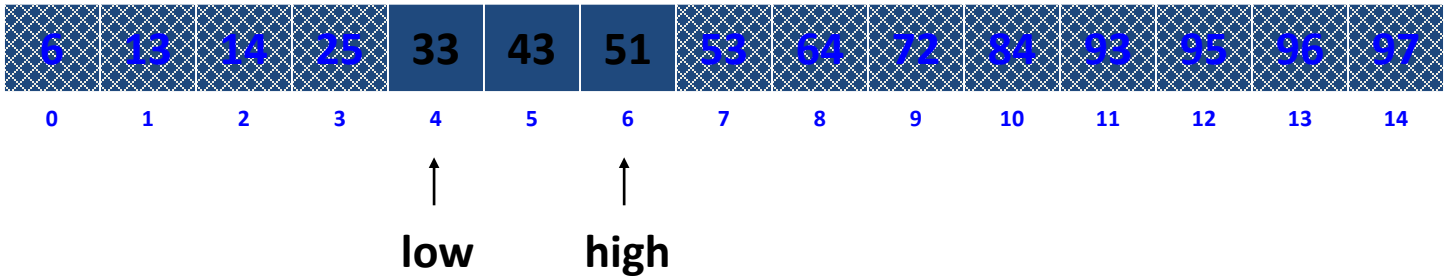


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

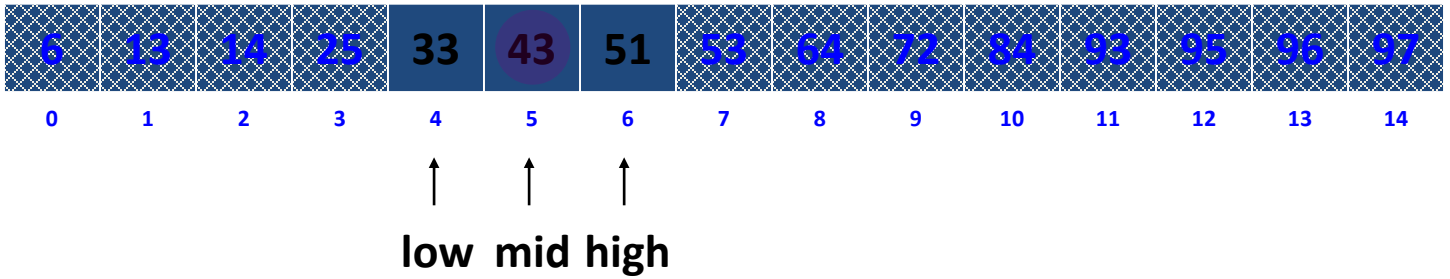


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

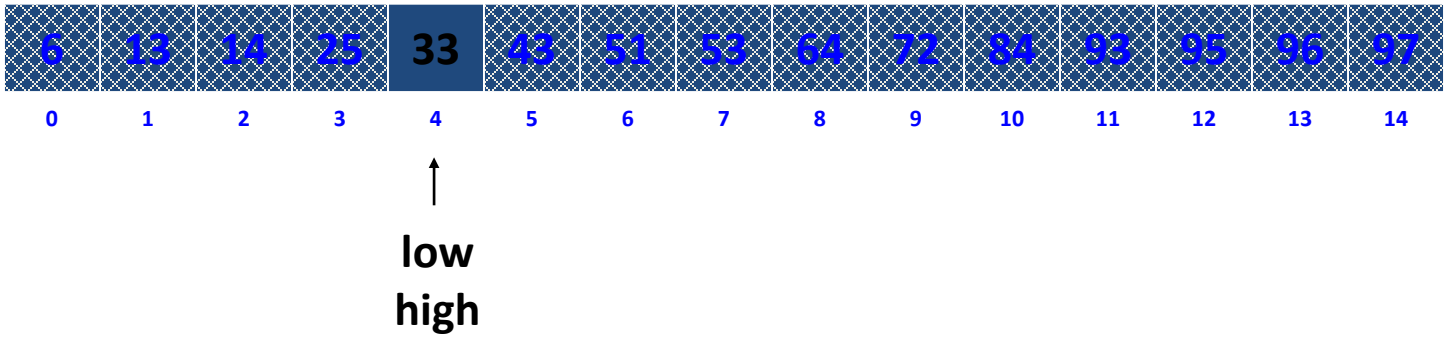


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

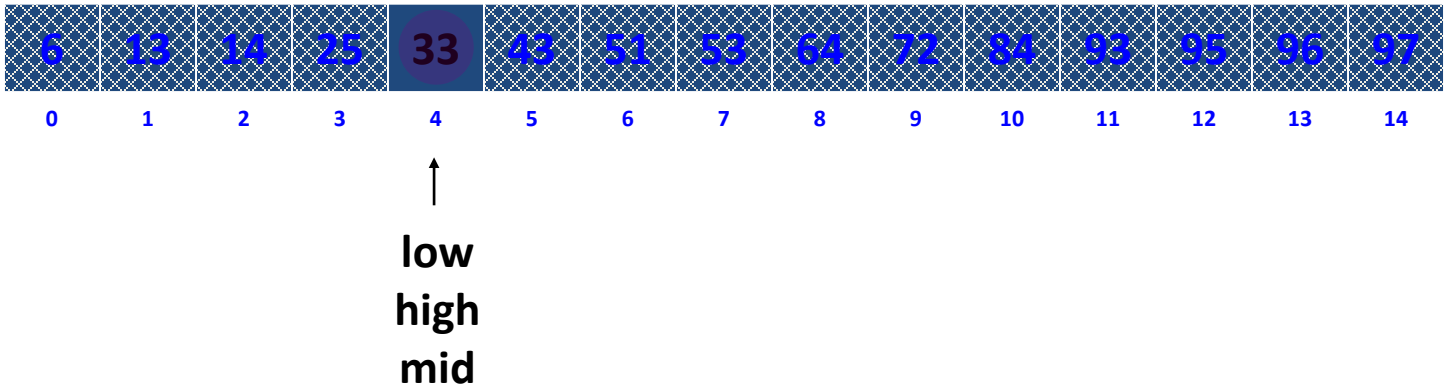


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.

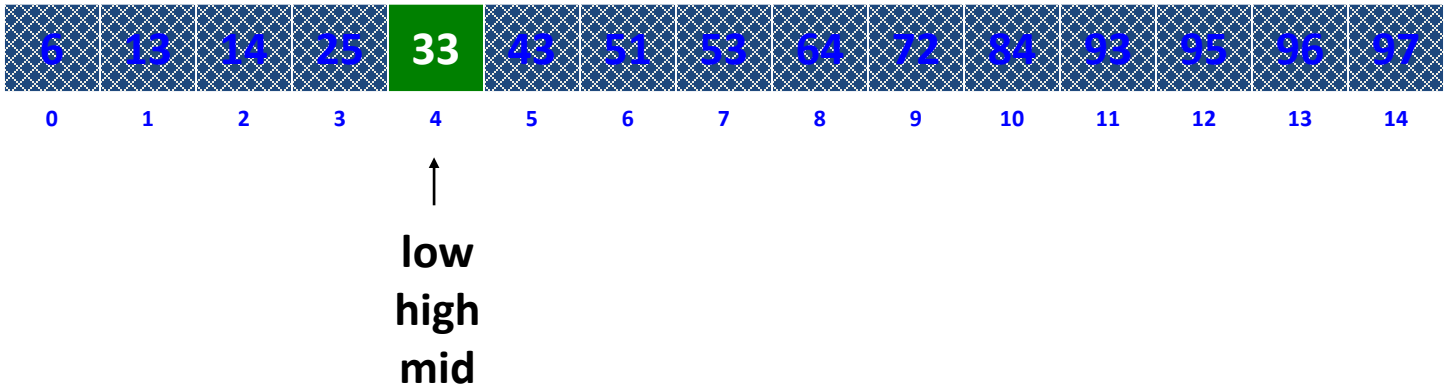


Binary Search

- Binary search. Given `value` and sorted array `a[]`, find index `i` such that `a[i]` = `value`, or report that no such index exists.

- Invariant. Algorithm maintains `a[low]` $\leq$ `value` $\leq$ `a[high]`.

- Ex. Binary search for 33.



Binary Search

```
low=0;  high=MAX-1;
while(low<=high){
      middle=(low+high)/2;
      if(key==array[middle]){
            printf("\nElement found at %d position",middle);
            break;
      }
      else if(key<array[middle])
         high=middle-1;
      else
         low=middle+1;
   }
```

Sorting

- Sorting
  - placing the data into a particular order such as ascending or descending

- Practical Examples
  - A bank sorts all checks by account number so that it can prepare individual bank statements at the end of each month
  - Telephone companies sort their lists of accounts by last name and, within that, by first name to make it easy to find phone numbers

Sorting Algorithms

- A sorting algorithm
  - An algorithm that puts elements of a list in a certain order
- The most used orders are numerical order

List of Various Sorting Algorithms

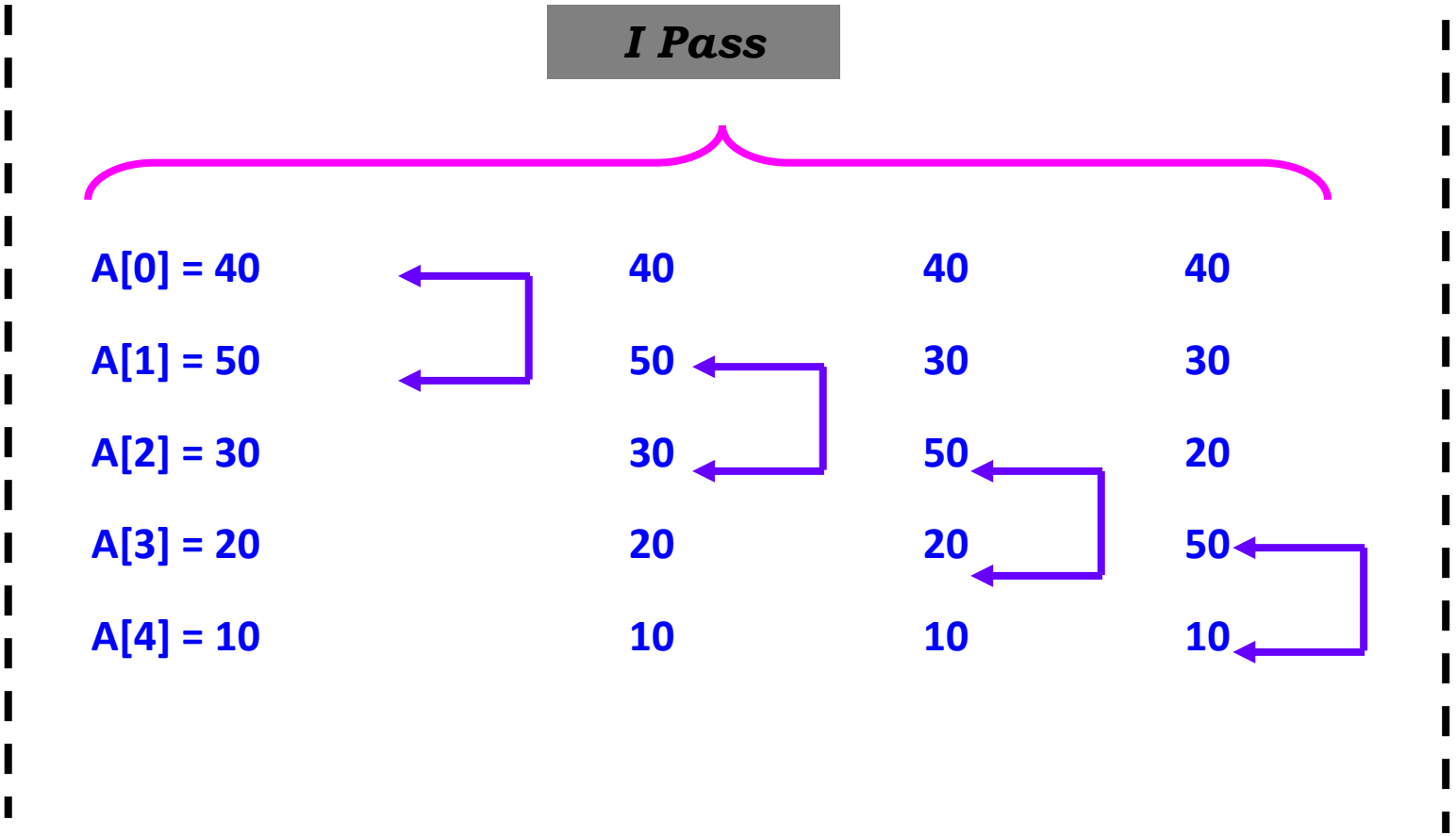
- Bubble Sort
- Selection Sort

Bubble Sort

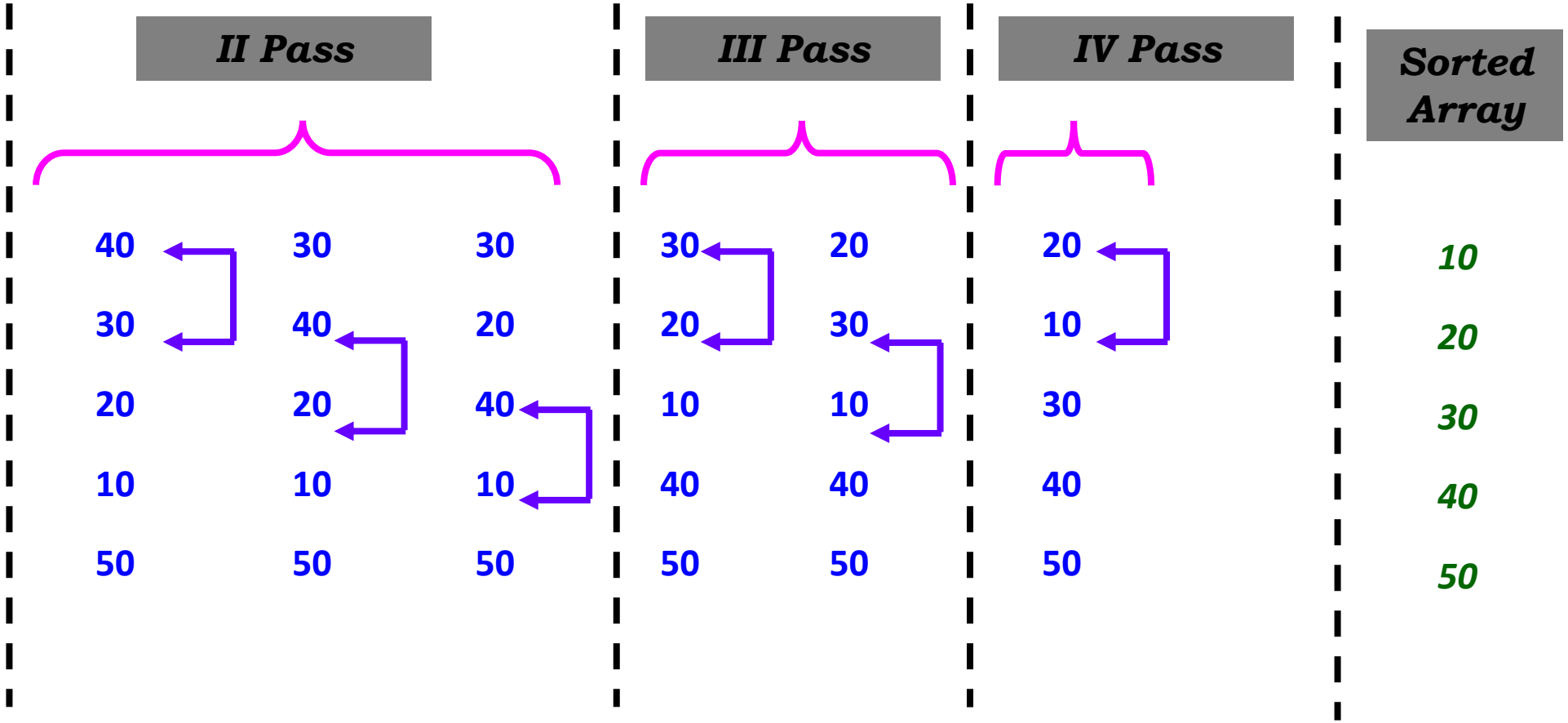
- Also known as exchange sort or sinking sort or comparison sort
  - the smaller values gradually “bubble” their way upward to the top of the array like air bubbles rising in water, while the larger values sink to the bottom of the array

- It works by repeatedly stepping through the list to be sorted, comparing two items at a time and swapping them if they are in the wrong order
- The pass through the list is repeated until no swaps are needed, which means the list is sorted

Trace of a Bubble Sort



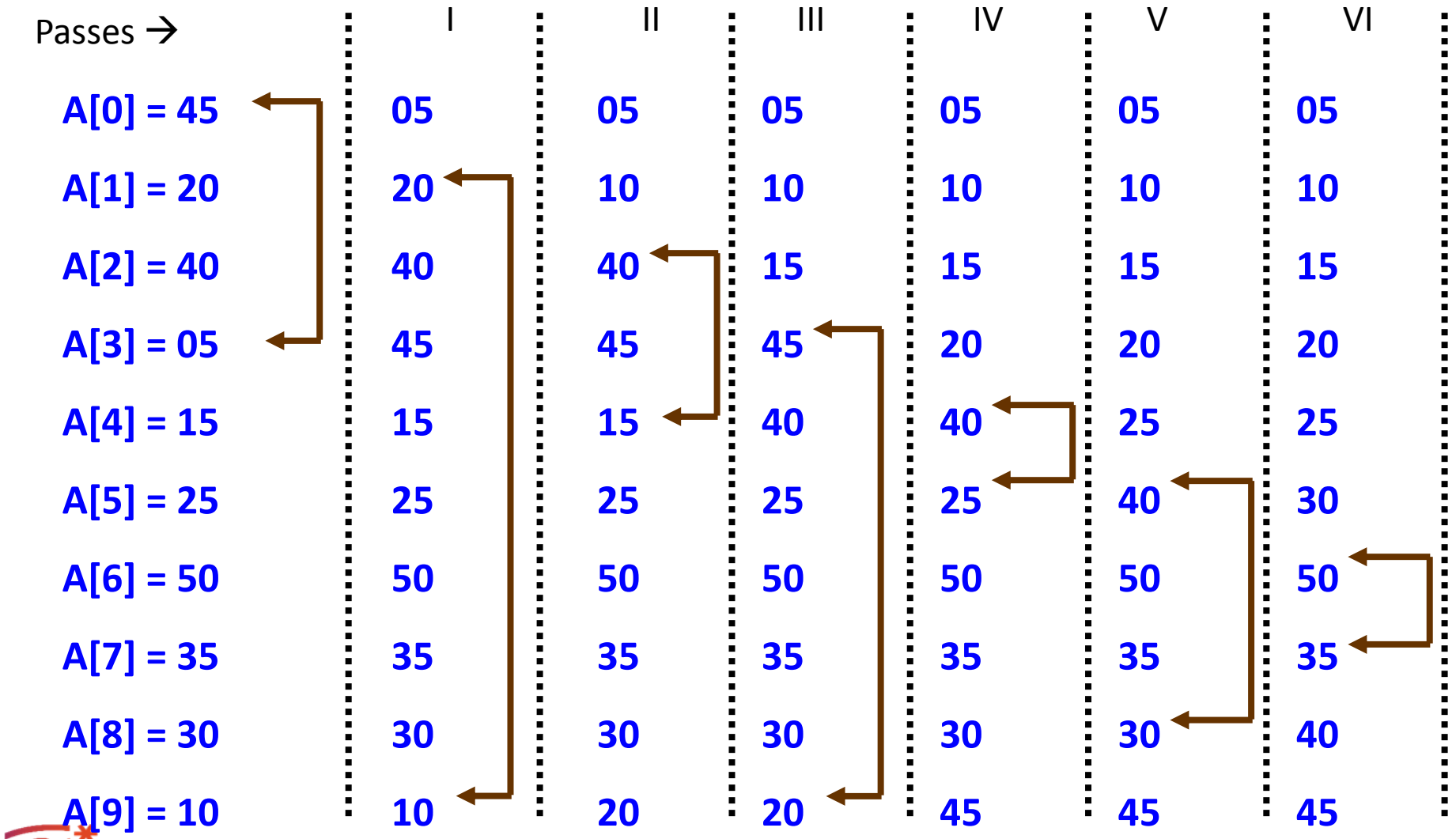
Trace of a Bubble Sort contd…



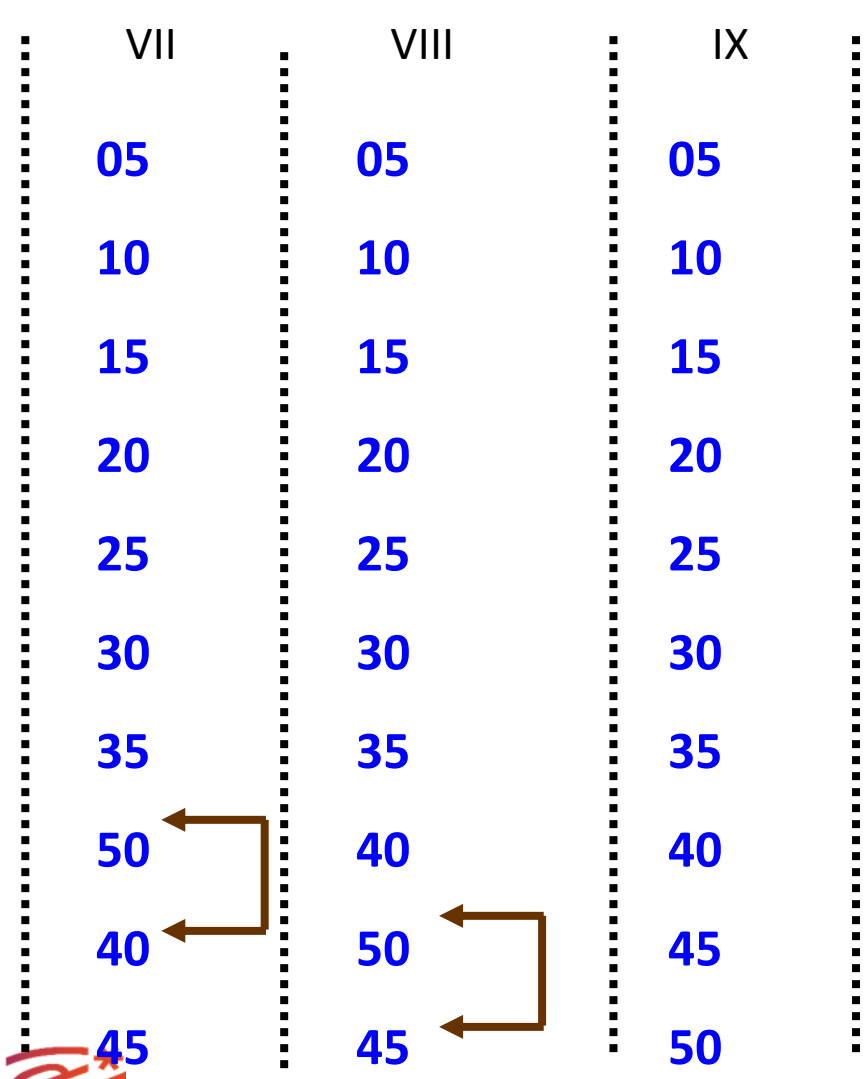
Selection Sort

- A sorting algorithm, specifically an in-place comparison sort
- Noted for its simplicity, and also has performance advantages over more complicated algorithms in certain situations
- It works as follows:
  1. Find the minimum value in the list
  2. Swap it with the value in the first position
  3. Repeat the steps above for remainder of the list (starting at the second position)

Trace of a Selection Sort



Trace of a Selection Sort contd….



Sorted Array

05
10
15
20
25
30
35
40
45
50

Summary

- Linear search is a search algorithm, also known as sequential search, that is suitable for searching a set of data for a particular value
- A binary search algorithm is a technique for finding a particular value in a linear array, by ruling out half of the data at each step .
- A sorting algorithm is an algorithm that puts elements of a list in a certain order
- Bubble sort is a simple sorting algorithm. It works by repeatedly stepping through the list to be sorted, comparing two items at a time and swapping them if they are in the wrong order
- Selection sort is a comparison sort in which the minimum element is exchanged with the first position repeatedly