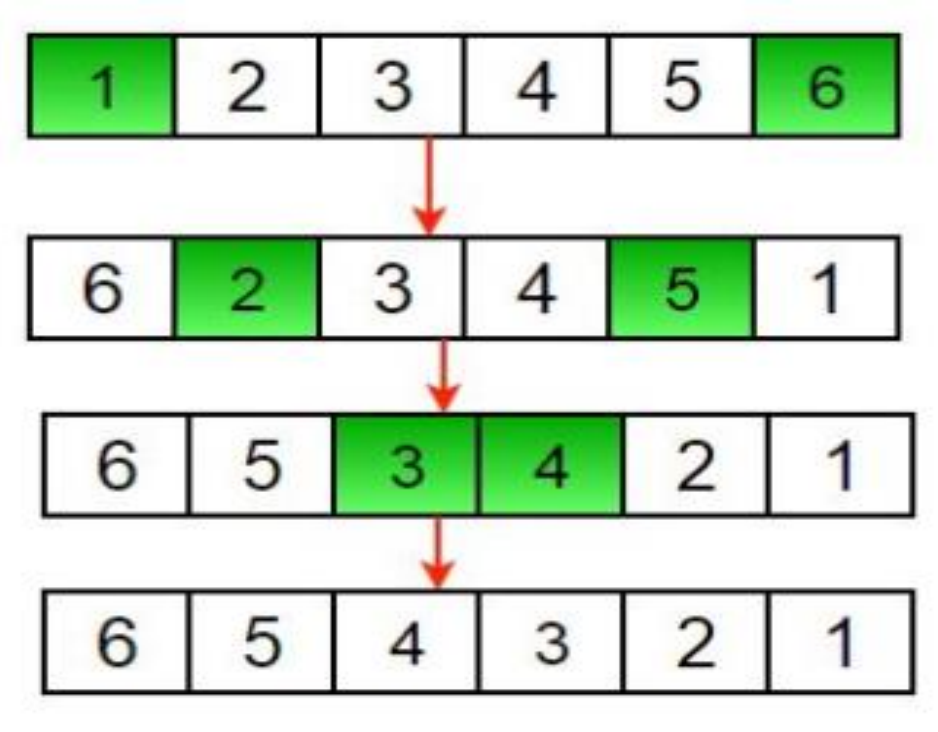
Programming Fundamentals

Aamina Batool

Reverse the elements



- int arr[100], i, j, total, temp;
- // get value of total from user and initialize the elements of array
- j = total – 1
- for (i=0; i<j ; i++, j--)
  - temp = arr[i]
  - arr[i] = arr[j]
  - arr[j] = temp;

Reverse even/odd indices

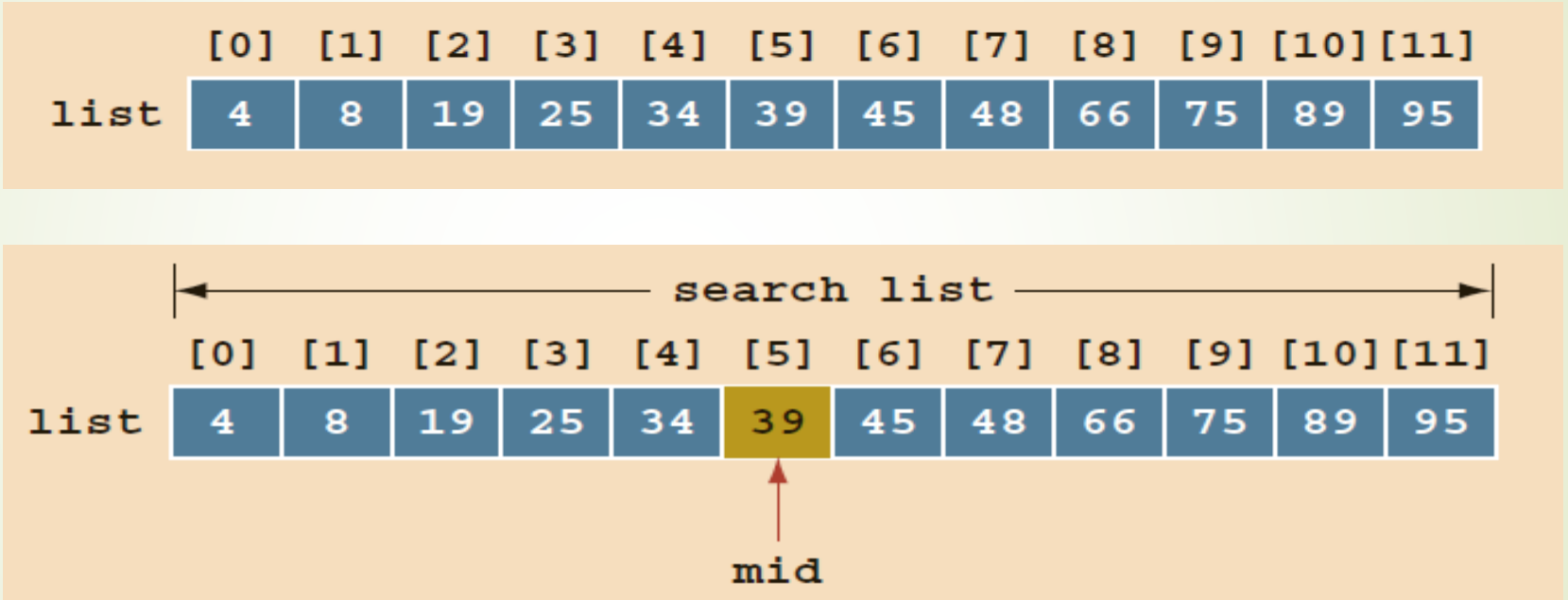
Searching an Array for a specific element

- Sequential/linear search
- Binary search

Sequential Search

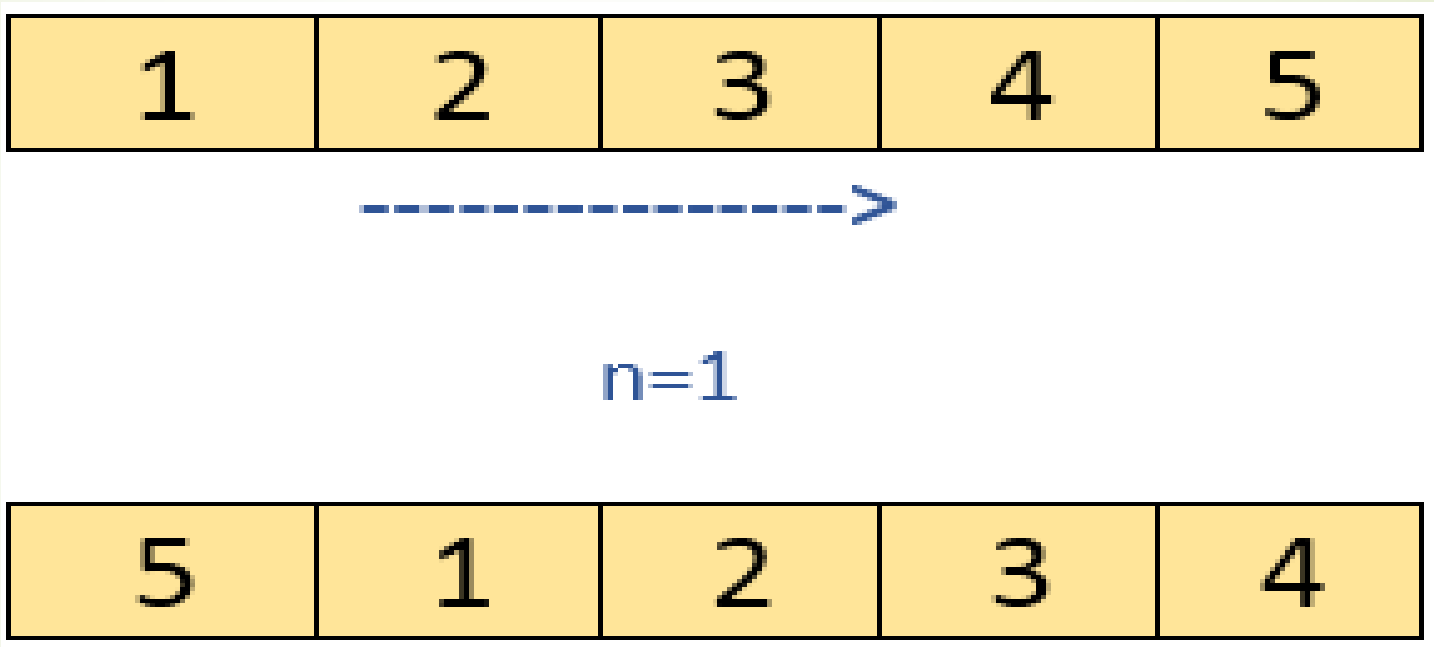
```
int loc;
bool found = false;
loc = 0;
while (loc < listLength && !found)
{
    if (list[loc] == searchItem)
        found = true;
    else
        loc++;
}
```

Binary Search



```
int first = 0;
int last = length - 1;
int mid;
bool found = false;
while (first <= last && !found)
{
    mid = (first + last) / 2;
    if (list[mid] == item)
        found = true;
    else if (list[mid] > item)
        last = mid - 1;
    else
        first = mid + 1;
}
```

Shifting/Rotating the Elements of Array



Insert/delete elements at start of array

- While initializing the array, we have to insert elements at the start of array only
- Delete an element from the start of the array and then shift all the elements back by one position

Equilibrium index

- Equilibrium index of an array is an index such that the sum of elements at lower indexes is equal to the sum of elements at higher indexes. For example, in an array A:
- ***Input**: A[] = {-7, 1, 5, 2, -4, 3, 0}*
  ***Output**: 3*
  3 is an equilibrium index, because:
  A[0] + A[1] + A[2] = A[4] + A[5] + A[6]
- ***Input**: A[] = {1, 2, 3}*
  ***Output**: -1*

- int i, j, leftSum, rightSum
- Array of size n
- For (i = 0 ; i < n; i++)
- {
  - leftSum = 0;
  - for (j=0; j <i ; j++}
    - leftSum += arr[j]
  - rightSum = 0;
  - for (j=i+1; j <n ; j++}
    - rightSum += arr[j]
  - If (leftSum == rightSum)
    - cout<< “The answer is “ << i << endl;
  - else
    - Cout<< -1 <<endl;
- }