K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Course: Data Structure

(Course Code : ENCS205)

UNIT-1: Foundations of Data Structures

School of Engineering & Technology
K.R. Mangalam University

SESSION 10:
Operations on Array

Recapitulation (Previous Session)

Quiz

Q1: What happens if you try to access an element at an index beyond the array size?

a) The program crashes.

b) The element at that index is set to null.

c) It returns an error.

d) It returns the last element of the array.

Q2: Which of the following is an example of a two-dimensional array?

a) int[] array;

b) int[][] matrix;

c) ArrayList<int[]> list;

d) HashMap<int, int> map.

Recapitulation (Previous Session cont..)

Q3: In C programming, how do you declare and initialize a static array of integers with 5 elements?

a) int array[5] = {1, 2, 3, 4, 5};

b) array<int> array = {1, 2, 3, 4, 5};

c) int array[] = {1, 2, 3, 4, 5};

d) Array arrayName = new Array(5);

Q4: What is the index of the last element in an array with size n?

a) n

b) n-1

c) n+1

Recapitulation (Previous Session)

Quiz (Answers)

A1: c) It returns an error.

A2: b) int[][] matrix;

A3: c) int array[] = {1, 2, 3, 4, 5};

A4: b) n-1

Array

- Contiguous Memory Allocation
- Fixed Size
- Index-Based Access
- Searching
- Homogeneous Elements
- Sequential Storage

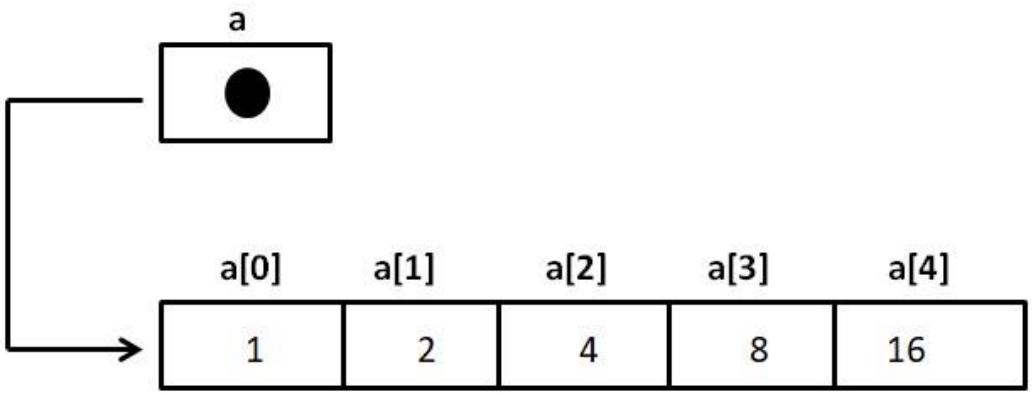


Fig. 1: Representation of array in memory

Array Operations

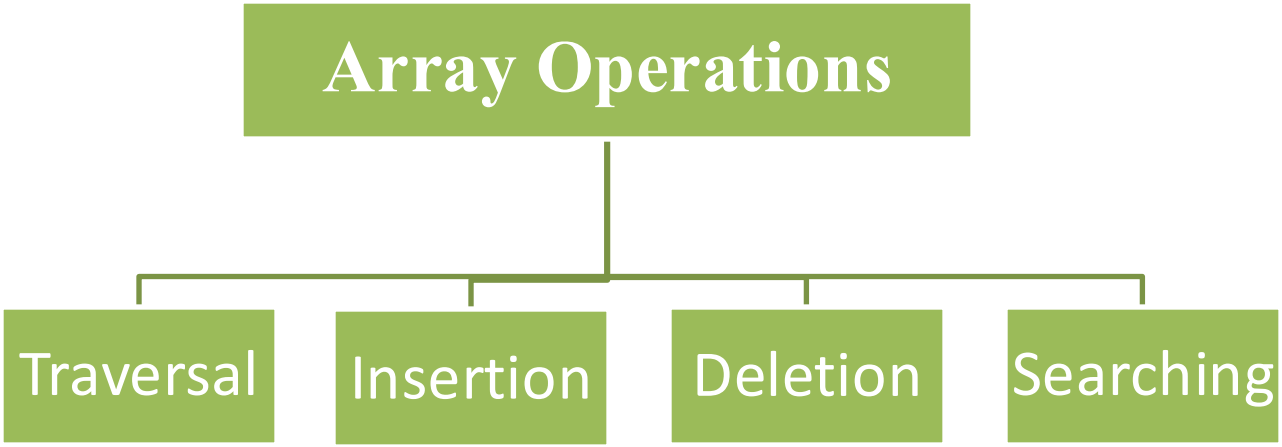
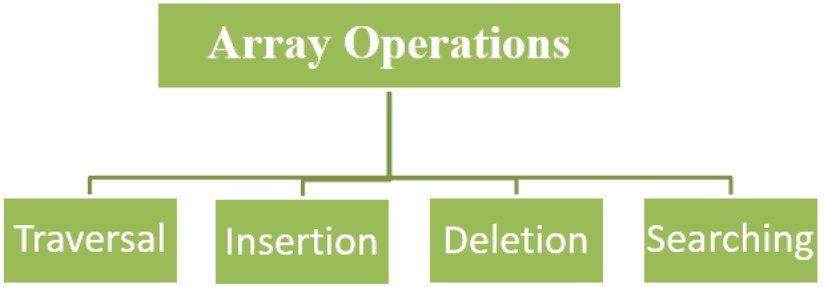


Fig. 2: Different Array Operations

Array Traversal

- Visiting all elements at once.




```
1  Start
2. Initialize an Array of certain size and datatype.
3. Initialize another variable ‘i’ with 0.
4. Print the ith value in the array and increment i.
5. Repeat Step 4 until the end of the array is reached.
6. End
```

Fig. 3: Algorithm of Array Traversal

Array Traversal (Program)

```c
int arr[] = { 1, 2, 3, 4, 5 };
int len = sizeof(arr) / sizeof(arr[0]);
// Traversing over arr[]
for (int i = 0; i < len; i++) {
    printf("%d ", arr[i]);
}
```

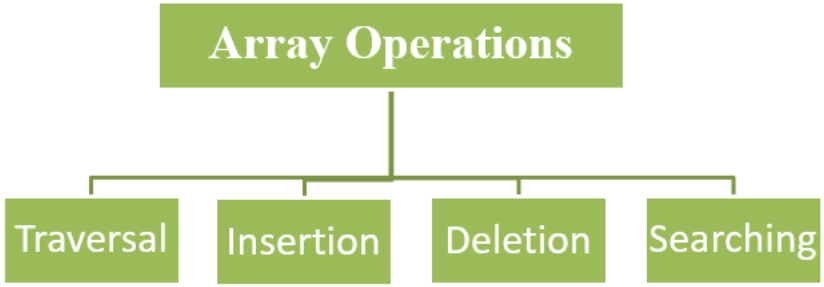
Fig. 4: Example of Array Traversal In language ‘C’

```python
import array
arr = array.array('i', [1, 2, 3, 4, 5])
# Traversing over arr[]
for x in arr:
    print(x, end=" ")
```

Fig. 5: Example of Array Traversal In language ‘Python’

Insertion in Array

- Insert one or more elements
- Insertion at any position in array




```
1. Start
2. Create an Array of a desired datatype and size.
3. Initialize a variable 'i' as 0.
4. Enter the element at ith index of the array.
5. Increment i by 1.
6. Repeat Steps 4 & 5 until the end of the array.
7. Stop
```

Fig. 6: Algorithm of Insertion in Array

Insertion in Array (Program)

```c
// Function to insert element
// at a specific position
void insertElement(int arr[], int n, int x, int pos)
{
    // shift elements to the right
    // which are on the right side of pos
    for (int i = n - 1; i >= pos; i--)
        arr[i + 1] = arr[i];

    arr[pos] = x;
}
```

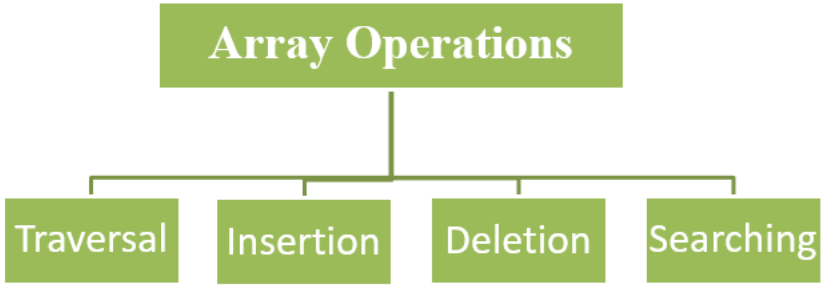
Fig. 7: Example of Insertion in language 'C'

```python
# python Program to Insert an element
# at a specific position in an Array


def insertElement(arr, n, x, pos):

    # shift elements to the right
    # which are on the right side of pos
    for i in range(n-1, pos-1, -1):
        arr[i + 1] = arr[i]

    arr[pos] = x
```

Fig. 8: Example of Insertion in language 'Python'

Deletion in Array

- Deletes element at any index.

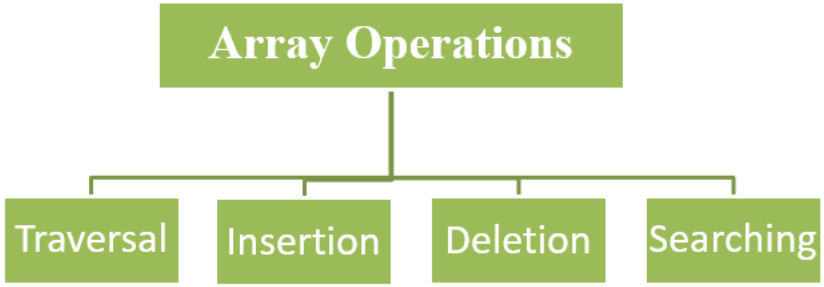



```
1. Start
2. Set J = K
3. Repeat steps 4 and 5 while J < N
4. Set LA[J] = LA[J + 1]
5. Set J = J+1
6. Set N = N-1
7. Stop
```

Fig. 9: Algorithm of Deletion in Array

Searching in Array

- Searching for an element.




```
1. Start
2. Set J = 0
3. Repeat steps 4 and 5 while J < N
4. IF LA[J] is equal ITEM THEN GOTO STEP 6
5. Set J = J +1
6. PRINT J, ITEM
7. Stop
```

Fig. 10: Algorithm of Deletion in Array

Searching in Array

```
// Function to implement search operation
int findElement(int arr[], int n, int key)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == key)
            return i;

    // If the key is not found
    return -1;
}
```

Fig. 11: Python program of Searching in Array

Time Complexity Analysis of operations an Array

Operation	Best Case	Average Case	Worst Case
Traversal	Ω(N)	θ(N)	O(N)
Insertion	Ω(1)	θ(N)	O(N)
Deletion	Ω(1)	θ(N)	O(N)
Searching	Ω(1)	θ(N)	O(N)

Space Complexity Analysis of operations an Array

Operation	Best Case	Average Case	Worst Case
Traversal	Ω(1)	θ(1)	O(1)
Insertion	Ω(1)	θ(N)	O(N)
Deletion	Ω(1)	θ(N)	O(N)
Searching	Ω(1)	θ(1)	O(1)

Applications of an Array Data Structure

- SearStoring and accessing data
- Sorting
- ching
- Matrices
- Stacks and queues
- Graphs
- Dynamic programming

Real-time Applications of an Array Data Structure

- Multimedia Applications
- Data Mining
- Robotics
- Real-time Monitoring and Control Systems
- Financial Analysis
- Scientific Computing
- Signal Processing

Practice Questions

Q1: Write a C++ function to find the maximum element in the array [8, 12, 6, 3, 15, 7, 1] . What is the maximum element?

Q 2: Write a C++ function to find the minimum element in the array [4, 9, 11, 2, 8, 6] . What is the minimum element?

Q 3: Write a C++ function to reverse the array [1, 2, 3, 4, 5]. What will be the resulting array?

Q 4: Write a C++ function to calculate the average of the elements in the array [10, 20, 30, 40, 50]. What is the average?

Practice Questions (Answers)

A1:
```
#include <iostream>
using namespace std;

int getMax(int arr[], int size) {
    int max = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    return max;
}
int main() {
    int arr[] = {8, 12, 6, 3, 15, 7, 1};
    int size = 7;

    int maxElement = getMax(arr, size);
    cout << "Maximum element: " << maxElement <<
endl;

    return 0;
}
```

Practice Questions (Answers)

A2:
```
#include <iostream>
using namespace std;

int getMin(int arr[], int size) {
    int min = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] < min) {
            min = arr[i];
        }
    }
    return min;
}
int main() {
    int arr[] = {4, 9, 11, 2, 8, 6};
    int size = 6;

    int minElement = getMin(arr, size);
    cout << "Minimum element: " << minElement << endl;
    return 0;
}
```

Practice Questions (Answers)

A3:
```
#include <iostream>
using namespace std;

void reverseArray(int arr[], int size) {
    int start = 0, end = size - 1;
    while (start < end) {
        swap(arr[start], arr[end]);
        start++;
        end--;
    }}
```

A4:
```
#include <iostream>
using namespace std;

int sumOfElements(int arr[], int size) {
    int sum = 0;
```

THANK YOU