Data Structure and Algorithm

DATA STRUCTURES ALGORITHMS

ARRAYS

Common Big O Algorithms

Name	Big O Notation	Example
Constant	O(1)	return true;
Logarithmic	O(log n)	binary search
Linear	O(n)	for or while loop
Quadratic	O(n^2)	loop within a loop
Exponential	O(c^n)	recursive calls over n and looping over c in the function
Factorial	O($n!$)	looping over n and recursive call in the loop to n-1

Arrays

IN COMPUTER SCIENCE, AN ARRAY IS A DATA STRUCTURE CONSISTING OF A COLLECTION OF ELEMENTS, EACH IDENTIFIED BY AT LEAST ONE ARRAY INDEX OR KEY.

Can contain multiple instances of a type

Can contain multiple instances of a type

Numeric indexing

Can contain multiple instances of a type
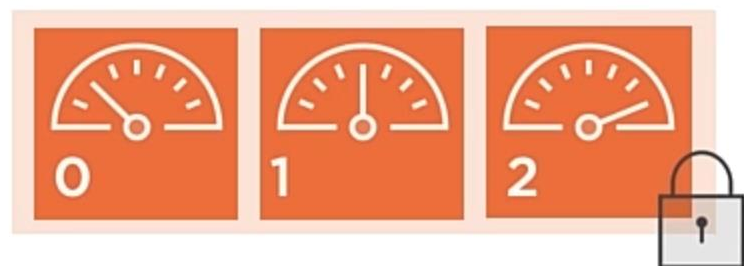
Numeric indexing

Access individual items

Can contain multiple instances of a type

Numeric indexing

Access individual items

Static or dynamic sizing

Can contain multiple instances of a type

Numeric indexing

Access individual items

Static or dynamic sizing

Fixed size once created

```
String[] cars;
```

Creating an Array

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Adding Data to an Array

```java
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);
```

Accessing Array Data

```java
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

for (int i = 0; i < cars.length; i++) {
    System.out.println(cars[i]);
}
```

Accessing Array Data

```java
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
cars[0] = "Opel";
System.out.println(cars[0]);
```

Updating Array Values

What is an Array?

An array is the simplest data structure where a collection of similar data elements takes place and each data element can be accessed directly by only using its index number.

Array Advantages
- ◦ Random access
- ◦ Easy sorting and iteration
- ◦ Replacement of multiple variables
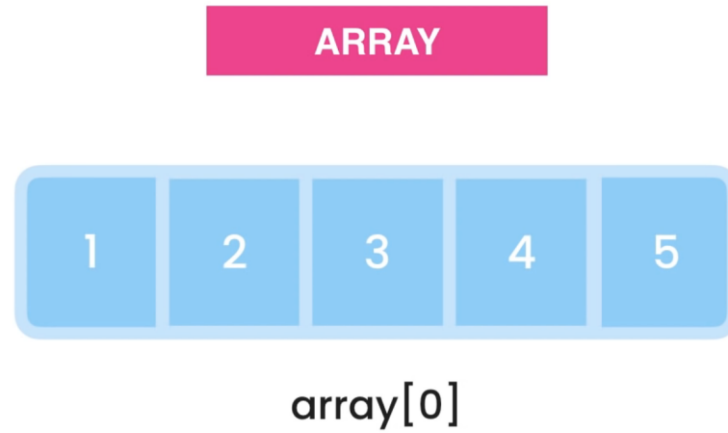
Array Disadvantages
- ◦ Size is fixed
- ◦ Difficult to insert and delete
- ◦ If capacity is more and occupancy less, most of the array gets wasted
- ◦ Needs contiguous memory to get allocated

Array Applications
- ◦ For storing information in a linear fashion
- ◦ Suitable for applications that require frequent searching

ARRAY

| 1 | 2 | 3 | 4 | 5 |
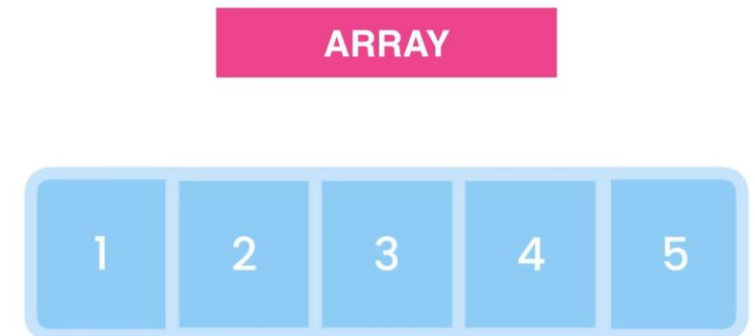
Accessing an Array

array[0]

- Accessing an array element by using its index is super fast

- but arrays have a fixed length and if you want to call / add / remove, we have to get resized

- this operation is costly as the input grows large

```java
package data_structures;

import java.util.*;

public class ArrayClass {
    public static void main (String[] args) {
        int[] priceOfPen= new int[5];
        Scanner in=new Scanner(System.in);
        for(int i=0;i<priceOfPen.length;i++)
            priceOfPen[i]=in.nextInt();

        for(int i=0;i<priceOfPen.length;i++)
            System.out.print(priceOfPen[i]+" ");
    }
}
```

Array Basic Operations

Traversal

- This operation is used to print the elements of the array.

Insertion

- It is used to add an element at a particular index.

Deletion

- It is used to delete an element from a particular index.

Search

- It is used to search an element using the given index or by the value.

Update

- It updates an element at a particular index.

Asymptotic Analysis of Algorithms

Resources

Operations
The number of times we need to perform some operations

Memory
How much memory is consumed by the algorithms

Others
Network transfer, compression ratios, disk usage

Questions?