ITS231: Data Structures and Algorithms Laboratory
School of Information, Computer, and Communication Technology
Sirindhorn International Institute of Technology

Lab 1: Recursion and Array

Objectives

- To review Java programming and concepts of Object-Oriented Program.
- To practice writing recursive programs
- To practice writing operation programs using Array datastrure

Reviews

A recursion is defining a repetition of procedures by making a call to itself.

A recursive program has two main cases:

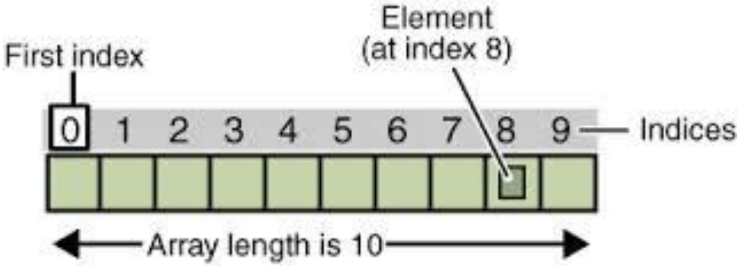
- **Base case**: tell when the calling-itself process should be stopped.
- **General case**: tell when the calling-itself process should continue and what needs to pass to the next call.

Example

```
int factorial (int n)
{    if(n==0)
     {    return 1;
     }
     else if( n > 0)
     {  return n* factorial (n-1);
      }
 }
```

Arrays is a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key.

- An array is stored so that the position of each element can be computed from its index tuble by a mathematical formula.
- The array size is fixed once you create it.
- It can store data at maximum equal to its size.
- Most computer languages start the index of Array at 0.



Array Operations

Operations using Array

Add

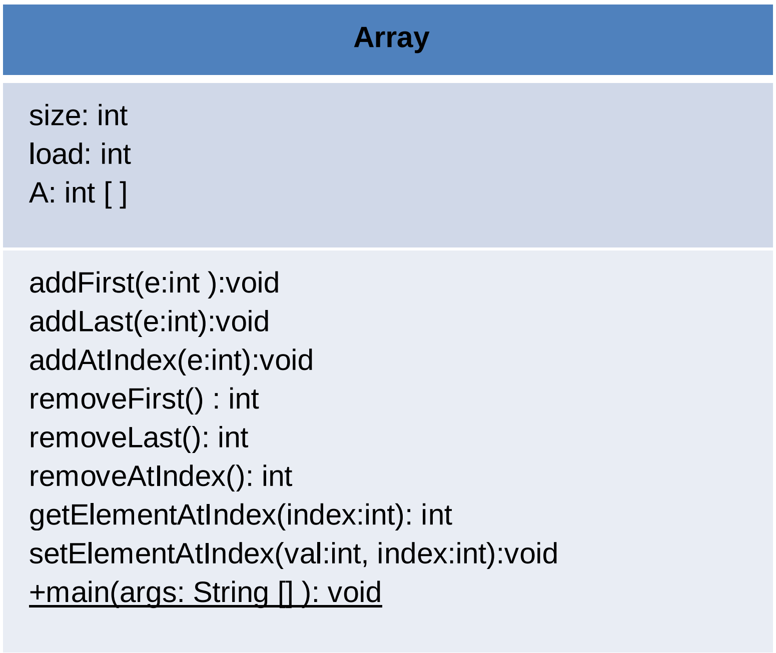
Add at first – insert the new element as the first element
Add at last – insert the new element next to the current last
Add at index – insert the new element at a specific index

Remove

Remove from first– remove the first element
Remove from last– remove the last element
Remove at index– remove an element at a specific index

Get value at index – get the element at a specific index
Set value at index– set a new value to the element a specific index

Array implementation



Array

size: int
load: int
A: int [ ]

addFirst(e:int):void
addLast(e:int):void
addAtIndex(e:int):void
removeFirst() : int
removeLast(): int
removeAtIndex(): int
getElementAtIndex(index:int): int
setElementAtIndex(val:int, index:int):void
+main(args: String []): void

Exercise 1 Complete the recursive program for the subtraction sum (`subsum`) for a positive input n.

$$subsum(n) = 1 - 2 + 3 - 4 + 5 - 6 + \cdots + (-1)^{n+1}n$$

In this problem you need to design the base case and the general case by yourself.

Exercise 2 Complete the recursive program `sumDigit` that computes the sum of the digits in an input which is assume to be a positive integer. In this problem you need to design the base case and the general case by yourself.

Exercise 3 Add methods in Array

a) Complete the method `addFirst(int val)` that inserts a new value to the first location of array
b) Complete the method `addLast(int val)` that inserts a new value to the last location of array
c) Complete the method `addAtIndex(int val, int index)` that inserts a new value val to a specific location index of array

Exercise 4 Remove methods in Array

a) Complete the method `int removeFirst()` that removes and returns a value in the first location of array
b) Complete the method `int removeLast()` that remove and returns a value in the last location of array
c) Complete the method `int removeAtIndex(int index)` that removes and returns a value in the location index of array

Exercise 5 Set and Get methods in Array

a) Complete the method `int getElementAtIndex (int index)` that get a value of array at location index.
b) Complete the method `void setElementAtIndex(int val, int index)` that sets value at index of A to val.